Relation Extraction from Web Contents with Linguistic and Web Features

Yulan Yan

Department of Information and Communication Engineering

University of Tokyo

A thesis submitted for the degree of

Doctor of Philosophy (Information Science & Technology)

March 2010

Acknowledgements

I wish to thank all my colleagues, friends and relatives who have given me help and encouragement during my work on this dissertation.

First and foremost, I am deeply grateful to my supervisor Prof. Mitsuru Ishizuka for his advice and warm support since I joined Ishizuka lab. This most important factor for the completion of this thesis was his understanding attitude towards my work and confidence in me.

I would like to express my warm thanks to Prof. Yutaka Matsuo for his scientific guidance, and invaluable comments through all my progress in this research. Furthermore, I would like to thank Dr. Naoaki Okazaki, Dr. Danushka Bollegala, Dr. Jie Yang, Dr. YingZi Jin, Mr. Haibo li, all the members of Ishizuka lab and Matsuo-gumi for their invaluable discussions and comments for my research work.

Many friends have encouraged me to finish my PhD course, high on this list are Dr. Zhenglu Yang, Ms. Yuzhen Li, Mr. Yongkun Wang, who deserve to receive my special thanks.

Finally, my thanks go to my family for their love and support through the years.

Abstract

With the advent of the Web and the explosion of available textual data, interest in techniques for machines to understand unstructured text has been growing. Recent attention to map textual content into a structured knowledge base through automatically harvesting semantic relations from unstructured text has encouraged Data Mining and Natural Language Processing researchers to develop algorithms for it. The relations can be defined in various levels regarding to their closeness to human understanding. One kind of relations is defined from the view of natural language understanding which is going through syntactic parsing towards semantic parsing. Many efforts have been focusing on how to represent sentence in structured representation. Identification of information from sentences and their arrangement in a structured format to be used in NLP and Web mining applications such as web searching and information extraction are expected. Another kind of relations is defined as binary relationships between named entities such as *birth_date*, *CEO* relations. Many recent efforts in this view have been focused on harvesting large scale of relational information from rom a local corpus or use the Web as corpus to build semantic repositories or ontologies for different applications such as question answering, semantic search.

In the first part of this thesis, we present a shallow semantic parser to add a new layer of semantic annotation of natural language sentences, facing the challenge of extracting a universal set of semantic or thematic relations covering various types of relations to represent sentence in a uniform structured representation. Our parser is based on the Concept Description Language for Natural Language (CDL.nl) which defines a set of semantic relations to describe the concept structure of text. In the second part, we propose several relation extraction methods to extract semantic relations from

Wikipedia. Currently frequent pattern mining-based methods and syntactic analysis-based methods are two types of leading methods for semantic relation extraction task. Using respective characteristics of Wikipedia articles and Web corpus, with a novel view on integrating syntactic analysis on Wikipedia text with redundancy information from the Web, we learn to discover and enhance relations in which a specified concept in Wikipedia participates with the complementary between the Web view and linguistic view. On the one hand, from the linguistic view, linguistic features (syntactic/dependency features) are generated from linguistic parsing on Wikipedia texts by abstracting away from different surface realizations of semantic relations. On the other hand, Web features (co-occurrence relational terms/textual patterns) are extracted from the Web corpus to provide frequency information by using a search engine.

In this thesis, we report evaluation results to illustrate the effectiveness and efficiency of our methods. For our shallow semantic parser, experiments on a manual dataset show that CDL.nl relations can be extracted with good performance. For our relation extraction systems from Wikipedia, evaluations demonstrate the superiority of the view combination over existing approaches. Fundamentally, we study the interrelated connection between linguistic and web views for semantic relation extraction. Our methods demonstrate how deep linguistic features contribute complementarily with Web features to the generation of various relations. Our study suggests an example to bridge the gap between Web mining technology and "deep" linguistic technology for information extraction tasks. It shows how "deep" linguistic features can be combined with features from the whole Web corpus to improve the performance of information extraction tasks. And we conclude that learning with linguistic features and Web features is advantageous comparing to only one view of features.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

With the dramatic increase in the amount of textual information available in digital archives and on the WWW, and since text documents convey valuable structured information, interest in techniques for automatically extracting information from text has been growing. For example, the medical literature contains information about new treatments for diseases. Similarly, news archives contain information useful to analysts tracking financial transactions, or to government agencies that monitor infectious disease outbreaks. All this information could be managed and queried more easily if represented in a structured form. This task is typically called information extraction.

In general, information extraction refers to automatic methods for creating a structured representation of selected information drawn from natural language text. There are two main topics with this task. 1):Recent years have been exhilarating ones for natural language understanding. The excitement and rapid advances that had characterized other language processing tasks such as speech recognition, part-of-speech tagging, and syntactic parsing have finally begun to appear in tasks in which understanding and semantics play a greater role. For example, there has been widespread commercial deployment of simple speech-based natural language understanding systems that answer questions about flight arrival times, give directions, report on bank balances, or perform simple financial transactions. More sophisticated research systems generate concise summaries of news articles, answer fact based questions, and recognize complex semantic and dialogue structure. 2): More specifically, information

extraction systems can identify particular types of entities (such as drug names) and relationships between entities (such as adverse interactions between medical drugs) in natural language text for storage and retrieval in a structured database [38]. Once created, the database can be used to answer specific questions quickly and precisely by retrieving answers instead of complete documents, for sophisticated query processing, for integration with relational databases, and for traditional data mining tasks.

But the challenges that lie ahead are still similar to the challenge that the field has faced since Winograd (1972)[83]: moving away from carefully hand-crafted, domain dependent systems toward robustness and domain-independence. This goal is not as far away as it once was, thanks to recent progresses in both natural language understanding and Web information extraction area. For example, the development of large semantic databases such as WordNet[31] and different levels of linguistic parsers, and progress in domain-independent machine learning algorithms for semantic information extraction. Based on all the existing technologies and algorithms, we choose to face the follow two challenges: 1) defining a universal set of semantic or thematic relations covering various types of semantic relationships between entities to format texts into structured representation; 2) combining linguistic analysis and large-scale Web information to support extracting semantic relationships between named entities automatically with as less as time-consuming human work.

To surmount the challenges of describing the concept structure of text into structured representation, we create a shallow semantic parser that used a new set of semantic relations of CDL.nl. The parser is developed as an intermediate phase in the progress to semantic parsing of natural language processing from dependency processing. CDL.nl relation set are defined to assume better coverage than those of Semantic Role Labeling - another shallow semantic parsing technology - to represent the concept structure of text. To overcoming the challenges of extracting semantic relational information automatically with as less as human work, we develop a semantic relation extraction system for automatically discovering interesting relations between entities from Wikipedia articles. In this system, relations are discovered by clustering pairs of co-occurring entities represented as vectors of context features with the combination of not only linguistic features by parsing Wikipedia texts using a "deep" linguistic parser, but also features from Web redundancy information by querying with the entities using a search engine. Relation clusters discov-

ered are then used for seeding a semi-supervised relation extraction algorithm to improve the coverage. We consider extracting semantic relations such as the CompanyHeadquarters(Organization:ORGANIZATION, Location:LOCATION) relation, CompanyCEO(Organization:ORGANIZATION, People:PEOPLE)

1.2 Thesis contributions

Our contributions can be summarized as the following.

With the first part of work,

- We develop a parser to add a new layer of semantic annotation of natural language sentences. Annotation of text with a deeper and wider semantic structure can expand the extent to which shallow semantic information can become useful in real semantic computing applications such as Information Extraction and Text Summarization.

- Our study shows an intermediate phase in the progress to semantic parsing of natural language processing from dependency processing.

- By modeling and leveraging lexical information separately from syntactic and dependency knowledge, our study also suggests an example of the flexibility of using kernel method to leverage diverse knowledge.

With the second part of work,

- Using characteristics of Wikipedia articles and the Web corpus respectively, with a novel view on integrating linguistic analysis on Wikipedia text with redundancy information from the Web, we propose an unsupervised relation extraction method for discovering and enhancing relations in which a specified concept in Wikipedia participates with the complementary between the Web view and linguistic view. From the Web view, related information between entity pairs are collected from the whole Web. From linguistic view, syntactic and dependency information are generated from appropriate Wikipedia sentences.

- Our study suggests an example to bridging the gap separating "deep" linguistic technology and redundant Web information for information extraction tasks. It shows how "deep" linguistic features can be combined with features from the whole Web corpus to improve the performance of information extraction tasks.
- Our experimental results reveal that relations can be extracted with good precision using linguistic features, while Web features from Web frequency information contribute greatly to the coverage of relation instances.
- We propose a multi-view learning approach for bootstrapping relationships between entities with the complementary between the Web view and linguistic view. We conclude that learning with linguistic features and Web features is advantageous comparing to only one view of features. Different from traditional multi-view learning approaches for relation extraction task, we filter view disagreement to deal with view corruption between linguistic features and Web features, only confident instances without view disagreement are used to bootstrap learning relations.

1.3 Organization of the Thesis

This thesis is organized as follows:

- In Chapter 2, we overview relation extraction task, and revisit related work systematically on relation extraction for linguistic parsing problem and semantic relation extraction problem from large scale of corpus. The relation extraction approaches can be either supervised or, unsupervised (or semi-supervised). First, we review the linguistic parsing techniques which are almost all supervised which are related to our first part of study - the development of a shallow semantic parser. Then, we review relation extraction technologies from large scale of Corpus with supervised, unsupervised and semi-supervised techniques, which are most closely related to the second part of our study - open relation extraction from Wikipedia.

- In Chapter 3, to surmount the challenges of describing the concept structure of text into structured representation, we present a shallow semantic parser to add a new layer of semantic annotation of natural language sentences based on the Concept Description Language for Natural Language (CDL.nl) which defines a set of semantic relations to describe the concept structure of text. The parsing task is a relation extraction process with two steps: relation detection and relation classification. Preliminary evaluation on a manual dataset shows that CDL.nl relations can be extracted with good performance.

- We then shift our focus to how to adapting linguistic parsing to scale information extraction from large collections facing the challenges of extracting semantic relational information automatically with as less as human work. Specifically, in Chapter 4, we present an unsupervised relation extraction system for discovering and enhancing relations in which a specified concept in Wikipedia participates by using respective characteristics of Wikipedia articles and Web corpus. Our performance study demonstrates that how deep linguistic patterns contribute complementarily with Web surface patterns to the generation of various relations.

- We study another unsupervised method by integrating frequency information from Web with linguistic analysis on Wikipedia texts in a multi-view co-clustering way for our open relation extraction from Wikipedia task in Chapter 5. The approach extends two co-clustering algorithms that are information theoretic co-clustering algorithm and self-taught clustering algorithm. We construct an integrated framework for relation extraction task consisting co-clustering functions for features and relations.

- In Chapter 6, we further study the open question of chapter 4: how to use the results of unsupervised clustering to harvest a large number of instances of these relations for Wikipedia concepts. We propose a multi-view learning approach for bootstrapping relationships between entities with the complementary between the Web view and linguistic view. The study indicates that bootstrap learning with linguistic features and Web features is advantageous comparing to learning with only one view of features.

- The thesis concludes in Chapter 7. I discuss potential future directions and conclude this thesis. In this thesis, we systematically studied two types of relation extraction: relation extraction for linguistic parsing and for semantic repository construction. We studied the first type of relation extraction by developing a shallow semantic parser to add a new layer of semantic annotation of natural language sentences. For the second type of relation extraction, we presented a serial of relation extraction methods for discovering and enhancing relations in which a specified concept in Wikipedia participates.

Chapter 2

Related Work on Relation Extraction

In this chapter, we will give an overview for relation extraction, and revisit the existing heuristic and algorithms.

2.1 Overview of Relation Extraction

The World Wide Web contains a significant amount of information expressed in natural language texts. Texts convey valuable structured information which could be managed and queried more easily if represented in a structured representation. The task of automatically constructing a structured representation of natural language text is typically called information extraction.

Attention for this task has encouraged Data Mining and Natural Language Processing researchers to develop algorithms for it in recent decades. Natural Language Processing researchers have been dedicating into natural language understanding which is going through syntactic parsing towards semantic parsing. Many efforts have been focusing on how to parse sentence in structured representation with nodes and relations. The structure representations can be defined in various levels regarding to their closeness to human understanding, with corresponding levels of relation types. The parsing of sentences and their arrangement in structured formats in different levels to be used in NLP and Web mining applications such as web searching and information extraction are expected. Recently following with the proposal of the Semantic Web, researchers on data mining have been interested in working on semantic resources with structured

information constructed from large scale text information. Many efforts have been dedicating into extracting concepts and relationships between concepts to create relational dababases to help people to retrieve semantic information automatically. Once created, the database can be used in different NLP and Web Mining applications, such as question answering to answer specific questions quickly and precisely by retrieving answers instead of complete documents, for sophisticated query processing, for integration with relational databases, and for traditional data mining tasks. Both tasks for NLP and data mining researchers can be treated as relation extraction task which is the subject of this thesis.

More specifically, relation extraction systems can identify particular types of entities and relationships between entities in natural language text for storage and retrieval in a structured database. A promising research direction is to automatically train the relation extraction systems and generate rules or extraction patterns for new tasks. With large amounts of text (annotated and otherwise) electronically available, the accuracy of the machine learning-based systems can rival that of manually-engineered systems. The machine learning approaches can be either supervised or, alternatively, unsupervised (or partially supervised). First, in Section 2.2, we review the linguistic parsing techniques which are almost all supervised which are related to our first part of study - development of a shallow semantic parser, in Section 2.3, we review relation extraction technologies from large scale of Corpus with supervised, unsupervised and semi-supervised techniques, which are most closely related to the second part of our study - open relation extraction from Wikipedia.

2.2 Relation Extraction for Linguistic Parsing

Recent years have been exhilarating ones for natural language understanding. The excitement and rapid advances that had characterized other language processing tasks such as speech recognition, part-of-speech tagging, and parsing have finally begun to appear in tasks in which understanding and semantics play a greater role. For example, there has been widespread commercial deployment of simple speech-based natural language understanding systems that answer questions about flight arrival times, give directions, report on bank balances, or perform simple financial transactions. More

sophisticated research systems generate concise summaries of news articles, answer fact based questions, and recognize complex semantic and dialogue structure.

Parsing is an important preprocessing step for many NLP applications and therefore of considerable practical interest. It is a complex task and as it is not straightforwardly mappable to a "classical" segmentation, classification or sequence prediction problem, it also poses theoretical challenges to machine learning researchers. During the last decade, much research has been done on data-driven parsing and performance has increased steadily. For training these parsers, syntactically annotated corpora (treebanks)[52] of thousands to tens of thousands of sentences are necessary; so initially, research has focused on English. During the last few years, however, treebanks for other languages have become available and some parsers have been applied to several different languages.

Tesnière (1959)[78] introduced the idea of a dependency tree (a "stemma" in his terminology), in which words stand in direct head-dependent relations, for representing the syntactic structure of a sentence. Hays (1964)[40] and Gaifman (1965)[35] studied the formal properties of projective dependency grammars, i.e. those where dependency links are not allowed to cross. Mel'čuk (1988)[54] describes a multistratal dependency grammar, i.e. one that distinguishes between several types of dependency relations (morphological, syntactic and semantic). Other theories related to dependency grammar are word grammar [45] and link grammar [73]. In fact, dependency parsing has been the subject of CoNLL shared tasks[10]; [39].

However, dependency parsing is not enough for machine to understand natural language texts. Automatic, accurate and wide-coverage techniques that can annotate naturally occurring text with semantic argument structure can play a key role in NLP applications such as Information Extraction[41], Question Answering[60] and Summarization. Semantic role labeling is the process of producing such a markup. When presented with a sentence, a parser should, for each predicate in the sentence, identify and label the predicate's semantic arguments. This process entails identifying groups of words in a sentence that represent these semantic arguments and assigning specific labels to them. In recent work, a number of researchers have cast this problem as a tagging problem and have applied various supervised machine learning techniques to it. Using correct syntactic parses it is possible to achieve accuracies rivaling human inter-annotator agreement. In fact, this has been the subject of two CoNLL shared tasks

([12, 13]. While all these systems perform quite well on the WSJ test data, they show significant performance degradation when applied to label test data that is different than the genre of the data that it was trained on. (Daniel Gildea and Daniel Jurafsky, 2002)[36] described a shallow semantic interpreter based on semantic roles that are less domain-specific than TO AIRPORT or JOINT VENTURE COMPANY. These roles are defined at the level of semantic frames of the type introduced by Fillmore (1976)[32], which describe abstract actions or relationships, along with their participates. Their paper describes an algorithm for identifying the semantic roles filled by constituents in a sentence.

But the challenges that lie ahead of researchers of natural language understanding are still similar to the challenge that the field has faced: defining a universal set of semantic or thematic relations covering various types of semantic relationships between entities to format texts into structured representation, to support constructing semantic structure automatically with as less as time-consuming human work in a robustness and domain-independence system.

2.3 Relation Extraction from Large Scale of Corpus

It is now almost universally acknowledged that stitching together the world's structured information and knowledge to answer semantically rich queries is one of the key challenges of computer science, and one that is likely to have tremendous impact on the world as a whole.

Wikipedia is the world's largest collaboratively edited source of encyclopedic knowledge in different languages. Wikipedia articles consist mostly of free text, but also contain different types of structured information, such as Infobox templates, categorization information, images, geo-coordinates, and links to external Web pages and links across different language editions of Wikipedia. Another important feature of Wikipedia is the presence of parallel articles in different languages. Certain pages represent direct translations as multilingual users build and maintain a parallel corpus in different languages. Recently, several projects such as YAGO/SOFIE, Kylin/KOG, and DBpedia, have successfully constructed large ontologies with relational information from Wikipedia articles based on different characteristics of Wikipedia.

In the second part of our work, we focus on the open Relation Extraction from Wikipedia task. Below we revisit two kinds of relation extraction methods which are mostly closed to our task from linguistic view and Web view respectively.

2.3.1 Supervised Relation Extraction from Linguistic View

The relation extraction task was first introduced as part of the Template Element task in MUC6 and then formulated as the Template Relation task in MUC7. Since then, many methods, such as feature-based ([47, 91]), tree kernel-based ([20, 88, 89]) and composite kernel-based ([89, 90]), have been proposed in literature.

For the feature-based methods, Kambhatla (2004)[47] employed Maximum Entropy models to combine diverse lexical, syntactic and semantic features in relation extraction, and achieved the F-measure of 52.8 on the 24 relation subtypes in the ACE RDC 2003 corpus. Zhou et al. (2005)[91] further systematically explored diverse features through a linear kernel and Support Vector Machines, and achieved the F-measures of 68.0 and 55.5 on the 5 relation types and the 24 relation subtypes in the ACE RDC 2003 corpus respectively. One problem with the feature-based methods is that they need extensive feature engineering. Another problem is that, although they can explore some structured information in the parse tree (e.g. [47] used the non-terminal path connecting the given two entities in a parse tree while Zhou et al. (2005) introduced additional chunking features to enhance the performance), it is found difficult to well preserve structured information in the parse trees using the feature-based methods. Zhou et al (2006) further improved the performance by exploring the commonality among related classes in a class hierarchy using hierarchical learning strategy.

As an alternative to the feature-based methods, the kernel-based methods (Haussler, 1999)[44] have been proposed to implicitly explore various features in a high dimensional space by employing a kernel to calculate the similarity between two objects directly. In particular, the kernel-based methods could be very effective at reducing the burden of feature engineering for structured objects in NLP researches, e.g. the tree structure in relation extraction. Zelenko et al. (2003) proposed a kernel between two parse trees, which recursively matches nodes from roots to leaves in a top-down manner. For each pair of matched nodes, a subsequence kernel on their child nodes is invoked. They achieved quite success on two simple relation extraction tasks. Culotta

and Sorensen (2004) extended this work to estimate similarity between augmented dependency trees and achieved the F-measure of 45.8 on the 5 relation types in the ACE RDC 2003 corpus. One problem with the above two tree kernels is that matched nodes must be at the same height and have the same path to the root node. Bunescu and Mooney (2005) proposed a shortest path dependency tree kernel, which just sums up the number of common word classes at each position in the two paths, and achieved the F-measure of 52.5 on the 5 relation types in the ACE RDC 2003 corpus. They argued that the information to model a relationship between two entities can be typically captured by the shortest path between them in the dependency graph. While the shortest path may not be able to well preserve structured dependency tree information, another problem with their kernel is that the two paths should have same length. This makes it suffer from the similar behavior with that of Culotta and Sorensen (2004): high precision but very low recall.

As the state-of-the-art tree kernel-based method, Zhang et al (2006) explored various structured feature spaces and used the convolution tree kernel over parse trees (Collins and Duffy 2001) to model syntactic structured information for relation extraction. They achieved the F-measures of 61.9 and 63.6 on the 5 relation types of the ACE RDC 2003 corpus and the 7 relation types of the ACE RDC 2004 corpus respectively without entity-related information while the Fmeasure on the 5 relation types in the ACE RDC 2003 corpus reached 68.7 when entity-related information was included in the parse tree. One problem with Collins and Duffy's convolution tree kernel is that the sub-trees involved in the tree kernel computation are context-free, that is, they do not consider the information outside the sub-trees. This is different from the tree kernel in Culota and Sorensen (2004), where the sub-trees involved in the tree kernel computation are context-sensitive (that is, with the path from the tree root node to the sub-tree root node in consideration). Zhang et al (2006) also showed that the widely-used Shortest Path-enclosed Tree (SPT) performed best. One problem with SPT is that it fails to capture the contextual information outside the shortest path, which is important for relation extraction in many cases. [92] randomly selected 100 positive training instances from the ACE RDC 2003 training corpus and showed that about 25% of the cases need contextual information outside the shortest path. [92] proposed a tree kernel with context sensitive structured parse tree information.

In order to integrate the advantages of feature based and tree kernel-based methods, some researchers have turned to composite kernel-based methods. Zhao and Grishman (2005) defined several feature based composite kernels to integrate diverse features for relation extraction and achieved the F-measure of 70.4 on the 7 relation types of the ACE RDC 2004 corpus. Zhang et al (2006) proposed two composite kernels to integrate a linear kernel and Collins and Duffy's convolution tree kernel. It achieved the F-measure of 70.9/57.2 on the 5 relation types/24 relation subtypes in the ACE RDC 2003 corpus and the F-measure of 72.1/63.6 on the 7 relation types/23 relation subtypes in the ACE RDC 2004 corpus. [92] proposed a context-sensitive convolution tree kernel and applied a composite kernel to combine the tree kernel and a state-of-the-art linear kernel for integrating both flat and structured features in relation extraction as well as validating their complementary nature. It achieved the F-measure of 74.1/59.6 on the 5 relation types/24 relation subtypes in the ACE RDC 2003 corpus and the F-measure of 75.8/66.0 on the 7 relation types/23 relation subtypes in the ACE RDC 2004 corpus.

2.3.2 Relation Extraction from Web View

The emergence of the WWW yields a dramatic increase of textual information. There is a great demand for organizing such textual data into structure to support machine processable. An important step in automating IE was the movement from knowledge-based systems to extractors learned from data. To that end, the goal of relation extraction techniques is to locate interesting entities and identify relations between them.

With the growth of the Web a massive quantity of documents, namely web pages, are freely available for (corpus-)linguistic studies. Web pages can be considered as a new kind of document, much more unpredictable and individualized than paper documents. However, Web pages are not only noisy at textual level. They also contain lots of physical noise. On a raw web page, i.e. a web page downloaded from the web without any pre-processing, many irregularities can be found, especially if the page has an HTML format. Unpredictable punctuation, typos, grammar mistakes, exotic names, extra-linguistic elements, such as HTML tags and code snippets, can make the use of NLP tools and automatic extraction of linguistic features hard. While today's Web search engines are useful tools for locating answers to many questions, collective

relevant information bout entities or entity pairs which scattered over many Web pages are feasible.

Keeping pace with progress in machine learning relations from the Web corpus, a diverse set of learning algorithms has been applied to this task, including support vector machines, hidden Markov models, conditional random fields and Markov logic networks. Nevertheless, the development of suitable training data for supervised RE requires substantial effort and expertise. Systems based on weakly-supervised learning, where a human provides a small number of seed relation instances that bootstrap learning over an unlabeled corpus, and unsupervised learning in an open way, where the system automatically finds and labels its own examples, further reduced the time required to develop IE systems. We now briefly survey these methods.

2.3.2.1 Weakly-Supervised Systems

DIPRE[9], Snowball[1] and KnowItAll[28] are among the most prominent projects of weakly-supervised relation extraction systems. They harness manually specified seed facts of a given relation (e.g., a small number of company-city pairs for a headquarter relation) to find textual patterns that could possibly express the relation, use statistics to identify the best patterns, and then find new facts from occurrences of these patterns.

Snowball[1] introduced strategies for generating patterns and extracting tuples from plain-text documents that required only a handful of training examples from users. At each iteration of the extraction process, Snowball evaluated the quality of these patterns and tuples without human intervention, and kept only the most reliable ones for the next iteration. The SatSnowball[93] proposed a statistical extraction framework called Statistical Snowball (StatSnowball), which is a bootstrapping system and can perform both traditional relation extraction and Open IE which can identify various types of relations without requiring pre-specifications. They focused on entity relation mining from the Web.

Bootstrapping methods [1, 9, 28] significantly reduce the number of training examples by iteratively discovering new extraction patterns and identifying entity relations with a small set of seeds, either target relation tuples [1] or general extraction templates [28]. However, the system [1] only generates patterns that are mainly based on keyword matching and its evaluation criteria are also specific to these strict high-precision

but low-recall patterns. Another bootstrapping system - KnowItAll [28] requires large numbers of search engine queries and web page downloads.

2.3.2.2 Open Relation Extraction

Typically, the target relation (e.g., seminar location) is given to the RE system as input along with hand-crafted extraction patterns or patterns learned from hand-labeled training examples [1, 9]. Such inputs are specific to the target relation. Shifting to a new relation requires a person to manually create new extraction patterns or specify new training examples. This manual labor scales linearly with the number of target relations. TextRunner[4] pursues the even more ambitious goal of extracting all instances of all meaningful relations from Web pages, a paradigm referred to as Open IE[29], which scales RE to the Web. An Open IE system extracts a diverse set of relational tuples without requiring any relation-specific human input. Open IE's extraction process is linear in the number of documents in the corpus, and constant in the number of relations. Open IE is ideally suited to corpora such as the Web, where the target relations are not known in advance, and their number is massive.

Sekine (2006) [70] developed a paradigm for ondemand information extraction in order to reduce the amount of effort involved when porting IE systems to new domains. Shinyama and Sekine [71] developed an unsupervised extraction process described as unrestricted relation discovery. Given a collection of documents, the system first clusters the articles using a bag-of-words document representation. Ideally, this step partitions the corpus into sets of articles believed to contain entities bearing similar relationships. Within each cluster, the system then performs named-entity recognition, reference resolution and linguistic parsing, and uses the output to form relational patterns used as features for an additional meta-clustering stage. Meta-clustering is computed in pairwise fashion over the set of entities found in the document cluster under consideration. Its output is a set of instances believed to participate in the same relationship, e.g. the relationship among a person, company and job-title involved in a hiring event. While the work of Shinyama and Sekine pursues the important goal of avoiding relation-specificity, it is unlikely to meet the scalability requirement necessary to process the Web. From a collection of 28,000 newspaper articles, Shinyama and Sekine were able to discover 101 relations, of which roughly 65% were judged

to be correct. For a corpus containing tens of thousands of documents, the relation discovery process was measured to take an average of 10 hours using a single 2.4GHz CPU with 4GB of memory.

Rosenfeld and Feldman (2007) [69] discover relationship instances by clustering entities appearing in similar contexts. Strategies were developed for discovery of multiple patterns for some specified lexical relationship (Pantel and Pennacchiotti, 2006) [66] and for unsupervised pattern ranking (Turney, 2006) [79]. Davidov et al. (2007) [23] use pattern clusters to define general relationships, but these are specific to a given concept. No study so far has proposed a method to define, discover and represent general relationships present in an arbitrary corpus. (Davidov and Rappoport, 2008) [24] presents an approach to extract pattern clusters from an untagged corpus. Each such cluster represents some unspecified lexical relationship. In this paper, they use these pattern clusters as the (only) source of machine learning features for a nominal relationship classification problem. Unlike the majority of current studies, they avoid using any other features that require some language-specific information or are devised for specific relationship types.

[43] introduced a method for discovering a relation by clustering pairs of co-occurring entities represented as vectors of context features. They used a simple representation of contexts; the features were words in sentences between the entities of the candidate pairs.

[79] presented an unsupervised algorithm for mining the Web for patterns expressing implicit semantic relations. Given a word pair, the output list of lexicon-syntactic patterns was ranked by pertinence, which showed how well each pattern expresses the relations between word pairs.

[23] proposed a method for unsupervised discovery of concept specific relations, requiring initial word seeds. That method used pattern clusters to define general relations, specific to a given concept. [24] presented an approach to discover and represent general relations present in an arbitrary corpus. That approach incorporated a fully unsupervised algorithm for pattern cluster discovery, which searches, clusters, and merges high-frequency patterns around randomly selected concepts.

The field of Unsupervised Relation Identification (URI)—the task of automatically discovering interesting relations between entities in large text corpora—was introduced

by [43]. Relations are discovered by clustering pairs of co-occurring entities represented as vectors of context features. [68] showed that the clusters discovered by URI are useful for seeding a semi-supervised relation extraction system. To compare different clustering algorithms, feature extraction and selection method, [69] presented a URI system that used surface patterns of two kinds: patterns that test two entities together and patterns that test either of two entities.

2.4 Characteristics of Wikipedia Articles

Wikipedia (www.wikipedia.org), a free encyclopedia on the web, has emerged as the world's largest encyclopedia. The term wiki indicates that information can be freely appended to the online encyclopedia by anyone who can access the site. Although it started from 2001, as of November 2009, the English Wikipedia contained more than 3 million articles. Because the encyclopedia is managed by the Wikipedia Foundation, an international non-profit organization, and because numerous collaborators in the world participate under some international projects, its articles are edited and developed continuously. For those reasons, its contents are believed to be quite reliable despite its openness.

Although Wikipedia contains an invaluable source of information, Wikipedia usage is currently limited to human readers [81] because the Wikipedia data format is quite difficult for machines to process: Wikipedia articles are written in natural language. In order to improve the usage of Wikipedia, it is necessary to represent Wikipedia's knowledge in the more formal format which supports machine-processable. This second part presents our work on structuring Wikipedia content by using relation extraction techniques. A relation is represented in form of a triple including a subject, a predicate and an object. For example, knowledge of the sentence: "Bill Gates is a founder of Microsoft Corp." can be encoded as (Microsoft Corp., founder, Bill Gates). Actually, such relations can then be transformed straightforwardly into RDF format[51], which in turn creates a machine-processable knowledge base for Semantic Web[5].

Wikipedia, unlike the whole Web corpus, has several characteristics that markedly facilitate information extraction. First, as an earlier report [37] explained, Wikipedia articles are much cleaner than typical Web pages. Because the quality is not so different from standard written English, we can use "deep" linguistic technologies, such as

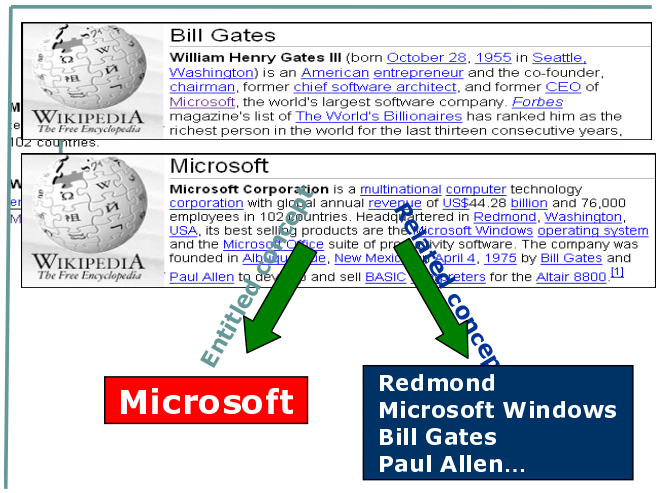

Figure 2.1: Characteristics of a Wikipedia article

syntactic or dependency parsing. Secondly, Wikipedia articles are heavily cross-linked, in a manner resembling cross-linking of the Web pages. [34] assumed that these links encode numerous interesting relations among concepts, and that they provide an important source of information in addition to the article texts. Fig. 2.1 illustrate these mentioned characteristics. We exploit the characteristics along with NLP techniques for this problem.

2.5 Relation Extraction Systems from Wikipedia

Recently, a number of projects have applied IE with specific focus on Wikipedia: DBpedia [2], work by Ponzetto et al. [67], Kylin/KOG [84, 85], and YAGO/SOFIE project [74, 75, 76]. While Ponzetto et al. focus on extracting a taxonomic hierarchy from Wikipedia, DBpedia and YAGO construct full-fledged ontologies from the semi-structured parts of Wikipedia (i.e., from infoboxes and the category system). They are not even tied to Wikipedia but can handle arbitrary Web pages and natural-language texts. Kylin goes beyond the IE in DBpedia and YAGO by extracting information not just from the infoboxes and categories, but also from the full text of the Wikipedia articles. KOG (Kylin Ontology Generator) builds on Kylin's output, unifies different attribute names, derives type signatures, and (like YAGO) maps the entities onto the WordNet taxonomy, using Markov Logic Networks. KOG builds on the class system

of YAGO and DBpedia (along with the entities in each class) to generate a taxonomy of classes. Both Kylin and KOG are customized and optimized for Wikipedia articles, while this paper aims at IE from arbitrary Web sources. SOFIE extends YAGO with information from the Web.

Wang et al. [82] have presented an approach called Positive-Only Relation Extraction (PORE). PORE is a holistic pattern matching approach, which has been implemented for relation-instance extraction from Wikipedia. Unlike the approach presented in this paper, PORE does not incorporate world knowledge, which would be necessary for ontology building and extension.

Our task to structure Wikipedia is also motivated by [81]. That work presents the design and implementation for an extension to Wikipedia called Semantic Wikipedia, which enables users to manually annotate knowledge to Wikipedia. The elements of the annotated knowledge include category, typed link, and attribute. Although category hierarchy already exists in Wikipedia, they introduce typed link and attribute as novel features that provide semantic information for the articles. Particularly, a typed link is defined as a relation between two pages, whereas attributes might be a descriptive value of a page, such as a number, date, etc. We automatically extract relations between entities in which each entity is discussed mainly in one article. Therefore, a typed link between articles is equivalent to a relation between corresponding entities. Our work can be considered as a step toward a Semantic Wikipedia. In other words, our work is to bridge the gap between the current Wikipedia and Semantic Wikipedia using automatic relation extraction techniques.

Unlike the web, Wikipedia articles contain few duplicated pieces of text that provide cues for relations between an entity pair. In other words, Wikipedia contents are not so abundant, which requires that all the texts be analyzed even if they have complex structure. As mentioned before, because Wikipedia articles are edited continuously by numerous collaborators, their content is believed to have high grammatical correctness compared to that of the web overall. Those assumptions enable us to exploit the syntactic structure of text, which is usually infeasible for ordinary web pages. We propose to make use of analysis of the linguistic (syntactic or dependency) structure of text. Put differently, analyzing the text at a syntactic or dependency level allows reduction of the variation of superficial text, which subsequently enables machines to recognize entity relations more accurately.

Chapter 3

A New Shallow Semantic Parser for Describing the Concept Structure of Text

3.1 Introduction

With the dramatic increase in the amount of textual information available in digital archives and on the WWW, interest in techniques for automatically extracting information from text has been growing. Identification of information from sentences and their arrangement in a structured format to be queried and used in semantic computing applications such as web searching and information extraction [16] are expected. Recently, much attention has been devoted to Semantic Role Labeling (SRL) of natural language text with a layer of semantic annotation having a predicate-argument structure, so-called shallow semantic parsing, which is becoming an important component in NLPs of various applications[60]. Currently, SRL is a well-defined task with a substantial body of work and comparative evaluation[13, 49]. Within the task of semantic role-labeling, high-performance systems have been developed using FrameNet[3] and PropBank[65] corpora, respectively, as training and testing materials.

Although Semantic Role Labeling specifically examines predicate-argument structure, towards the goal of putting the whole sentence into a semantic structural form, Yokoi et al. (2005)[86] presented a descriptive language named Concept Description Language for Natural Language (CDL.nl), which is part of the realization of spirits

of the work "semantic information processing"[56]. In fact, CDL.nl defines a set of semantic relations to form the semantic structure of natural language sentences in a graphical representation. They record semantic relationships showing how each meaningful entity (nominal, verbal, adjectival, adverbial) relates semantically to another entity. It connects all meaningful entities into a unified graphical representation, not only predicate-argument related entities.

Consequently, using the CDL.nl relation set, the task of structure annotation becomes a relation-extraction process that is divisible into two steps: relation detection, which is detecting entity pairs for which each there exists a meaningful relationship; and relation classification, which is labeling of each detected entity pair with a specific relation. For CDL.nl relation extraction, the challenge we must confront is that not only the relation detection step is more difficult than a classification problem as in semantic role labeling, but also that classification of a wide variation of CDL.nl relation types is harder than that of only predicate-argument roles. In this thesis, we describe a hybrid approach using two different methods for each step: first, based on dependency analysis, a rule-based method is presented for relation detection; secondly, a kernel-based classification method is presented to assign a CDL.nl relation to each detected entity pair by leveraging different levels of syntactic analysis.

Our contributions can be summarized as the following.

- We develop a parser to add a new layer of semantic annotation of natural language sentences. Annotation of text with a deeper and wider semantic structure can expand the extent to which shallow semantic information can become useful in real semantic computing applications such as Information Extraction and Text Summarization.

- Our study shows an intermediate phase in the progress to semantic parsing of natural language processing from dependency processing.

- By modeling and leveraging lexical information separately from syntactic and dependency knowledge, our study also suggests an example of the flexibility of using kernel method to leverage diverse knowledge.

The remainder of this chapter is organized as follows. Section 3.2 explains the background in the semantic role labeling domain relating to semantic roles in FrameNet,

PropBank, and semantic role labeling tasks. Section 3.3 introduces the CDL.nl relation set and specifies its importance and challenges. Section 3.4 proposes our hybrid method for relation extraction. Section 3.5 reports our preliminary experimental results and our observations. We conclude our work in Section 3.6.

3.2 Background

During the last few years, corpora with semantic role annotation and automatic annotation systems have received much attention. Three corpora are available for developing and testing predicate-argument annotation–FrameNet[3], PropBank[65], and NomBank[55]. Semantic role labeling is the process of assigning a simple WHO did WHAT to WHOM, WHEN, WHERE, WHY, HOW, etc. structure to sentences in text. In this section, we specifically address semantic role labeling systems which are based on FrameNet and PropBank.

3.2.1 FrameNet Semantic Roles

The Berkeley FrameNet project, started in 1998, is primarily a corpus-based lexicon-building project that documents the links between lexical items and their semantic frame(s). Its starting point is the observation that words can be grouped into semantic classes, so-called 'frames', a schematic representation of situations involving various participants, props, and other conceptual roles. Each frame has a set of predicates (nouns, verbs, or adjectives), which introduce the frame. For each semantic frame, it defines a set of semantic roles called **frame elements**, which are shared by all predicates of the frame.

For example, the frame **Intentionally_create** shown in Fig. 3.1 is denominated using a set of semantically related predicates such as **verbs** *make* and *found*, **nouns** *creation* and *generation*, and is defined as follows: The *Creator* creates a new entity, the *Created_entity*, possibly out of *Components*.

The roles defined for this frame include **core roles** *Created_entity* and *Creator*, **non-core roles** *Co_participant*, *Components*, and so on. A number of hand-annotated examples from the **Intentionally_create** frame are included below to give a flavor of the FrameNet database:

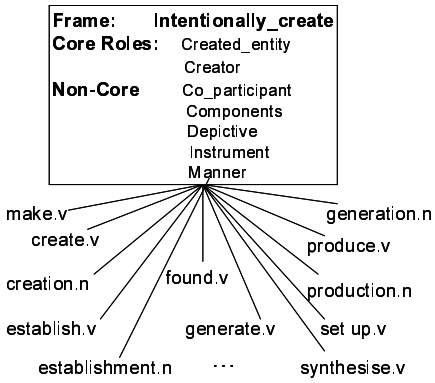

Figure 3.1: Sample frame from FrameNet lexicon

(1) [*Creator* Breeders] have **ESTABLISHED** [*Created_entity* their own intelligence network] [*Purpose* in a bid to combat the crime].
(2) A [*Created_entity* US$100,000,000 EFTA development fund for Yugoslavia] was **ESTABLISHED** [*Time* in April 1990].
(3) The Supreme Soviet decisions were based on findings by [*Created_entity* an official commission] **SET UP** [*Time* in May 1989] [*Purpose* to examine the investigators' work].
(4) [*Created_entity* A memorial fund] has been **SET UP** [*Place* in his native village] [*Purpose* to build a monument to one of Ulster's less remembered legion of fighting men].
(5) The steam produced by this process is in turn used to drive [*Cause* large turbines which] **GENERATE** [*Created_entity* electricity] [*Manner* in exactly the same way as in any other conventional power station].
(6) [*Creator* Caroline Gordon], [*Creator* who] has **FOUNDED** [*Created_entity* Beaver Recruitment Brokers] [*Co_participant* with Joan Tannian], said that the market was flooded with recruitment agencies which created confusion among companies with positions to fill.
(7) [*Creator* The NIERC] was **ESTABLISHED** [*Time* in October 1985] [*Role* as an independent research center funded jointly by the ESRC, the Northern Ireland government and private industry].

Semantic role labeling processing

Based on the FrameNet annotation system, given a crude sentence, the standard role labeling process goes through (1) identifies all predicates, (2) disambiguates the frame for each predicate, and (3) labels the roles of arguments related to the predicate based on the frame definition.

*Bill Gates is an American entrepreneur and the [***Role** *chairman] of [***Jurisdiction** *Microsoft], [***Created_entity** *the software company] [***Creator** *he] founded [***Co_participant** *with Paul Allen] [***Place** *in Albuquerque, New Mexico] [***time** *on April 4, 1975].*

Above is an example showing how to annotate a sentence using FrameNet roles. It is apparent that it annotates only predicate-argument roles and only for predicates "chairman" and "found", not for "entrepreneur" which is not encoded in any frame. Since FrameNet lists only 10197 lexical units, comparing to 207016 word-sense pairs in WordNet 3.0, it also has the limitation that the roles are frame-specific, and only predicates from certain predetermined semantic frames can be annotated.

3.2.2 PropBank Semantic Roles

The FrameNet labels are rather rich in information. However, they might not always be transparent for users and annotators. The Proposition Bank (PropBank) lexicon was put forward first in 2000 to facilitate annotation, and later evolved into a resource in its own right, with the intention of adding a layer of semantic annotation to the Penn English TreeBank with verb-argument structure. Therefore, the advantage of the PropBank approach is that, using neutral labels, less effort is required from annotators to assign them.

The Proposition Bank aims to provide a broad-coverage hand annotated corpus of such phenomena, enabling the development of better domain-independent language understanding systems, and the quantitative study of how and why these syntactic alternations take place. Because of the difficulty in defining a universal set of semantic or thematic roles covering all types of predicates, PropBank defines semantic roles on a verb-by-verb basis. PropBank is constructed following a "bottom-up" strategy: starting from various senses of a word, a frame-file is created for every verb. Such a frame-file therefore contains all possible senses of the verb plus a set of example

sentences that illustrate the context in which the verb can occur. For each sense of the verb, a role set and example sentences are available.

Because of the difficulty of defining a universal set of semantic or thematic roles covering all types of predicates, PropBank defines semantic roles on a verb by verb basis. An individual verb's semantic arguments are numbered, beginning with 0. For a particular verb, the verb-specific numbered roles covered by PropBank are the following:

Numbered arguments (*A0-A5*, *AA*): Semantic arguments of an individual verb are numbered beginning with 0. For a particular verb, *Arg0* is generally the argument exhibiting features of a prototypical Agent whereas *Arg1* is a prototypical Patient or Theme. The meaning of each argument label depends on the usage of the verb in question.

As examples of verb-specific numbered roles, we give entries for the verbs *accept* and *kick* below. These examples are taken from the guidelines presented to the annotators.

(8) Frameset accept.01 "take willingly"
Arg0: Acceptor
Arg1: Thing accepted
Arg2: Accepted-from
Arg3: Attribute
Example:[*Arg0* He] [*ArgM-MOD* would][*ArgM-NEG* n't] accept [*Arg1* anything of value] [*Arg2* from those he was writing about].

(9) Frameset kick.01 "drive or impel with the foot"
Arg0: Kicker
Arg1: Thing kicked
Arg2: Instrument (defaults to foot)
Example1: [*ArgM-DIS* But] [*Arg0* two big New York banksi] seem [*Arg0* $trace_i$] to have kicked [*Arg1* those chances] [*ArgM-DIR* away], [*ArgM-TMP* for the moment], [*Arg2* with the embarrassing failure of Citicorp and Chase Manhattan Corp. to deliver $7.2 billion in bank financing for a leveraged buy-out of United Airlines parent UAL Corp].
Example2: [*Arg0* Johni] tried [*Arg0* $trace_i$] to kick [*Arg1* the football], but Mary pulled

Table 3.1: Subtypes of the ArgM modifier tag

LOC: location	CAU: cause	PRD (secondary predication)
TMP: time	PRP: purpose	DIS: discourse connectives
MNR: manner	ADV: general-purpose	NEG: negation marker
DIR: direction	MOD: modal verb	DIS (discourse particle and clause)
EXT: extent		

it away at the last moment.

(10) Frameset edge.01 "move slightly"
Arg0: causer of motion Arg1: thing in motion Arg2: distance moved
Arg3: start point Arg4: end point Arg5: direction
Example: [*Arg0* Revenue] edged [*Arg5* up] [*Arg2-EXT* 3.4%] [*Arg4* to $904 million] [*Arg3* from $874 million] [*ArgM-TMP* in last year's third quarter].

In addition to verb-specific numbered roles, PropBank defines several more general roles that can apply to any verb. In addition to the semantic roles described in the rolesets, verbs can take any of a set of general, adjunct-like arguments (ArgMs), distinguished by one of the function tags shown in Table 3.1. **Adjuncts (*AM-*)**: General arguments that any verb might take optionally. There are 13 types of adjuncts such as *AM-ADV*(general-purpose), *AM-TMP*(temporal).

Each verb's roles are numbered, as in the following occurrences of the verb offer from Propbank data:
(11) ...[*Arg0* the company] to ... offer [*Arg1* a 15% to 20% stake] [*Arg2* to the public].
(12) ... [*Arg0* Sotheby's] ... offered [*Arg2* the Dorrance heirs] [*Arg1* a money-back guarantee]
(13) ... [*Arg1* an amendment] offered [*Arg0* by Rep. Peter DeFazio] ...
(14) ... [*Arg2* Subcontractors] will be offered [*Arg1* a settlement] ...

Semantic role labeling processing

Based on the PropBank annotation system, given a sentence, the role-labeling process goes through (1) identifies each verbal predicate and (2) labels its arguments.

*Bill Gates is an American entrepreneur and the chairman of Microsoft, [*ARG1 *the software company] [*ARG0 *he] [*rel *founded] [*AM-MAN *with Paul Allen] [*AM-LOC *in Albuquerque, New Mexico] [*AM-TMP *on April 4, 1975.]*

Shown above is an example portraying how to annotate a sentence using PropBank roles. It is readily apparent that PropBank specifically examines verb predicate–argument roles.

3.2.3 Semantic Role Labeling Tasks

Gildea and Jurafsky[36] (2002) presented the first semantic role labeling system to apply a statistical learning technique based on FrameNet data. They describe a discriminative model for determining the most probable role for a constituent given the predicate with its frame. This task has been the subject of a previous Senseval task (Automatic Semantic Role Labeling)[49] and two shared tasks on semantic role labeling in the Conference on Natural Language Learning (2004&2005)[13].

Systems contributed to the Senseval shared task were evaluated to meet the same objectives as the Gildea and Jurafsky study using the FrameNet data. In Senseval-3, two different cases of automatic labeling of semantic roles were considered. The Unrestricted Case requires systems to assign semantic roles to the test sentences for which the boundaries of each role were given and the predicates identified. The Restricted Case requires systems to (i) recognize the boundaries of semantic roles for each evaluated frame as well as to (ii) assign a label to it. Eight teams participated in the task, with a total of 20 runs for two cases. The average precision over all Unrestricted Case runs is 80.3% and the average recall is 75.7%. The average precision over all Restricted Case runs is 59.5% and the average recall is 48.1%, which is notably lower than the first case, underscoring the additional difficulty of identifying the frame element boundaries.

Using CoNLL-2004, 2005, a shared task evaluated SRL systems based on the PropBank corpus. Given a sentence with several target verbs marked, a semantic role labeling system develops a machine-learning system to recognize and label the arguments of each verb predicate. In all, 19 systems participated in the CoNLL-2005 shared task. They approached the task in several ways, using different learning components

and labeling strategies with different types of linguistic features, providing a comparative description and results. Evaluation is performed on a collection of unseen test sentences that are marked with target verbs and which contain only predicted input annotations; the best results in the shared task almost reached F1 at 80% in the WSJ test set, and almost 78% in the combined test.

3.3 CDL.nl Semantic Relation Extraction Task

This thesis follows the approach of seeking for the models for language processing like those in human understanding. Particularly, we propose to use binary relations between pairs of concepts or pairs of entities for knowledge representation. In the first part of this thesis, we introduce Concept Description Language (CDL), an artificial language which is designed to represent knowledge and semantics from all data formats. Yokoi et al. (2005)[86] presented Concept Description Language for Natural Language (CDL.nl), which is used to describe the semantic/concept structure of text as a core component of W3C Common Web Language[1]. Its motivation is to share knowledge between computers and human. As a result, it enables computers to process information semantically and in turn to provide greater satisfaction of users' needs.

Different from existing dependency parsing, which represent the grammatical dependency structure of text, it is used to describe the semantic structure of plain text in graphical form. Similar to the aforementioned approaches for Natural Language Understanding, the two basic elements for describing the structure are Entity and Relation, where the element Entity is used to represent a constituent of sentences with a head word. Therefore, a set of entities and relations forms a concept structure of underlying knowledge. A set of relations[2] is defined to represent the meaning of the relationships between a pair of entities. The entity which heads the relation is called the head entity; the other one is the tail entity. A lexicon named UNLKB is used to organize entities for CDL.nl according to their semantic behaviors. they are based on their participant relations. More details about the lexicon are presented in Section 3.4.2.3.

[1] http://www.w3.org/2005/Incubator/cwl/

[2] http://www.miv.t.u-tokyo.ac.jp/mem/yyan/CDLnl/

Obviously, one of the important tasks is to create CDL representations automatically. Our work is to focus on CDL.nl, a CDL version for natural language. The task of transforming CDL representations from text can be divided into two subtasks: that of identifying pairs of entities between which it is likely to have a relation and that of identifying the relation label for the pairs. In this thesis, we focus on the second problem. That means, we develop label classifiers given pairs of entities between which there exists a relation. Although CDL can represent information and knowledge described by any kind of natural languages, the classifiers we present in this thesis processes for English text.

3.3.1 CDL.nl - CDL for Natural Language

Institute of Semantic Computing[1] is developing an artificial language to enable knowledge sharing between computer and human, which is called Concept Description Language (CDL) [86]. While the existing computer languages are designed under the viewpoint of computer mechanisms such as computation mechanism, data structure, program structure, and so on, CDL is designed under the viewpoint of human and of content which human uses. In other words, processing resources of CDL are contents and semantics. As Semantic Web aims at enabling semantic processing on web data, CDL goal is to extend the media it supports, which may include textual data, visual data and acoustic data. As a result, in order to enable the knowledge sharing between human society and computers, CDL should play the following roles of an intermediary:

- Intermediate language among media: to intermediate among various kind of data such as multiple languages, between natural language and formal language (mathematical language, programming language, etc.), and between various media, and so on.
- Intermediate language from shallow semantic processing on the conceptual level to deeper semantic processing/knowledge processing.
- Intermediate language between syntactic document structure processing (XML) and semantic document structure processing.

[1] http://www.instsec.org/

CDL.nl is a version of CDL for representing knowledge and concept structure encoded in natural languages, which is derived from Universal Networking Language (UNL)[80]. CDL.nl graph also consists of the following basic elements:

- **Entity:** Entity of CDL.nl expresses a concept from natural language. A concept of CDL.nl can be a class, an instance, a single concept or a compound concept. In natural language text, it can be expressed by a word, a phrase or a sentence. Entity can be considered as an instance of Universal Word (UW) of UNL in which it realizes a concept corresponding to a UW in UNL Knowledge Base[1] by providing contextual information of the text for the UW.

- **Relation:** Defines relationships between pairs of Entities. CDL uses directed binary relations to describe objectivity information of sentences. The types of relationships are differentiated by labels. For example, the relations in Fig. 3.2 indicate that the agent initiating the action "report" is "John", the target of the action "report" is "Alice", and so on.

- **Attribute:** Describes the subjectivity of sentences including time with respect to the speaker (past, present, future), speaker's view of aspect (begin, continue, complete...), speaker's view of reference (specific, non-specific...), speaker's focus (emphasis, theme, title...), speaker's attitudes (confirmation, exclamation...) and speaker's view point (ability, admire...).

Fig. 3.2 shows a CDL's sample graph of the concept structure of the sentence: "John reported to Alice that he bought a computer yesterday."

3.3.2 CDL.nl Semantic Relation Set

With similar objectives to those of PropBank to add a layer of semantic annotation on natural language sentences, but different from roles in PropBank, where role semantics depends on the verb and verb usage, or verb sense in a sentence, CDL.nl predefines a set of neural semantic relations covering different types of predicates. Furthermore, additional information for distinguishing similar relations is also described. For example, the definition of *aoj* (nominal entity with attribute) contains two parts:

[1] http://www.undl.org/unlsys/uw/unlkb.htm

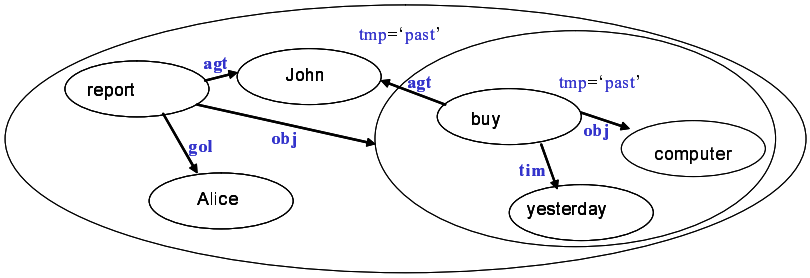

Figure 3.2: CDL's graph of a sample sentence

Definition: *aoj* indicates a nominal thing that is in a state or has an attribute.
Differences between related relations: A thing with an attribute differs from *mod* in that *mod* gives some restriction of the concept that is being analyzed, whereas *aoj* signifies a thing of a state or characteristic.
Example: for the short sentence " Leaves are green", there is a relation typed as *aoj* between green and leaves, so the machine can understand that "leaves" here have the attribute "green".

Facing the challenge of defining a universal set of semantic or thematic relations covering various types of semantic relationships between entities, CDL.nl defines a set of semantic relations containing 44 relation types which are organized into three groups: (Please refer to Appendix A for a full list of CDL's relations)

- **Intra-event relation**: Relations defining case roles, which are divided into the six abstract relations of *QuasiAgent*, *QuasiObject*, *QuasiInstrument*, *QuasiPlace*, *QuasiState*, and *QuasiTime*. Furthermore, each abstract relation includes several concrete relations which express concrete semantic information. For example, *QuasiAgent* contains five semantic relations: *agt* (agent), *aoj* (thing with attribute), *cag* (co-agent), *cao* (co-thing with attribute), *ptn* (partner). To illustrate the advantage of these subset relations, we take the *cag* (co-agent) as an example: in the sentence "John walks with Mary", "Mary" is the co-agent of event "walks". Consequently, we know both the facts that "John walks" and "Mary walks".

- **Inter-entity relations**: In addition to event-specific numbered roles, CDL.nl defines 13 more general relation types that can apply to different types of the

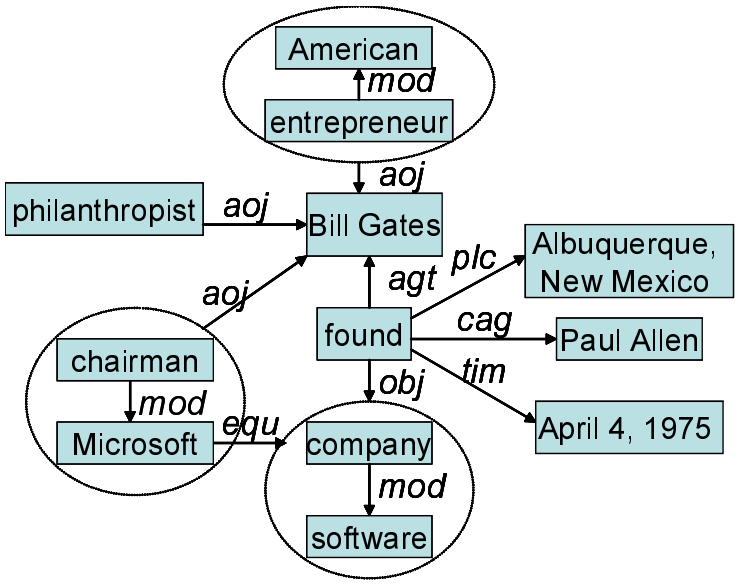

Figure 3.3: The graphic structure of sentence "*Bill Gates is an American entrepreneur, philanthropist and chairman of Microsoft, the software company he founded with Paul Allen in Albuquerque, New Mexico on April 4, 1975.*"

head entity. As the definition of relation type *pur* (purpose) shows, in addition to the action entity, NominalEntity can activate the *pur* relation. Other inter-entity relations are *seq* (sequence), *equ* (equivalent), etc.

- **Qualification relations**: Relations representing qualification relationships between modified entity and modifier entity. There are nine qualification relations, collectively containing *mod* (modification), *pos* (possessor), and *qua* (quantity). This subset of relations is important to describe an entity with myriad properties.

Compared to FrameNet and PropBank, the CDL.nl relation set is useful to annotate not only facts in sentences about WHO did WHAT to WHOM or with WHOM, WHEN, WHERE, WHY, and HOW, but also What has WHICH properties, and so on. A directed graph, in which Entity is designated as a node and Relation is regarded as an arc, is useful to represent the semantic structure. An Entity is classifiable into an elemental entity and a composite entity. Composite Entity is a hyper node which contains the structure of Entity and Relation within it. It is a syntactically phrase, clause

or sub-sentence. Unlike a hyper node in graphical theory, however, nodes inside and outside the Composite Entity might be mutually linked by a direct arc.

Fig. 3.3 illustrates an example showing the graphical structure annotated using CDL.nl relations. Comparing to annotation with FrameNet and PropBank, it supports our idea that, with the CDL.nl relation set, plain sentences can be annotated not only using predicate-argument relations, but also that between each pair of entities, there exists a meaningful relationship, such as the *equ* (equivalent) relation between the entities "Microsoft" and "the software company", which shows that both refer to the same object, and *aoj* (thing with attribute) relation between "American entrepreneur" and "Gates", showing that "Gates" has the attribute of "American entrepreneur".

3.3.3 Challenges of Automatic CDL.nl Relation Extraction

The task of structure annotation with the CDL.nl relation set can be seen as a relation extraction process that is divisible into two steps: relation detection, or the detection of entity pairs between each pair for which there exists a meaningful relationship; and relation classification, or the labeling of each detected entity pair with a specific relation.

Considering the first step, semantic role detection in SRL systems involves only classifying each syntactic element in a sentence as either a semantic argument or a non-argument by assigning a predicate, so that it is a binary-classification problem. However, the task of detecting a CDL.nl relation is not strictly a classification problem; conceptually, the system must consider all possible subsequences (i.e. consecutive words) pairs in a sentence. In this respect, the detection of dependency relations resembles that of our relation detection task. As evident from the CoNLL-X shared task on dependency parsing [10], two dominant models are currently used often for data-driven dependency parsing. The first is "all-pairs" approach [53], by which every possible arc is considered in the construction of the optimal parse. The second is the "stepwise" approach [64], by which the optimal parse is built stepwise and where the subset of possible arcs that is considered depends on previous decisions. Clearly, the "all-pairs" approach requires exponential time in its worst case. Furthermore, although the "stepwise" approach builds a parse depending on prior decisions, our task of CDL.nl relations annotated in sentences are mutually independent. For that reason, the

challenge of our first step of relation extraction is that we need an efficient method that is adequate for independent relation detection considering all possible subsequences.

For the second step, although semantic role classification involves classification of each semantic argument identified into a specific semantic role, our relation classification task involves assigning a specific CDL.nl relation to each detected entity pair to form the graphical structure of the sentence. The challenges are: (1), we must consider all 44 relation types simultaneously; (2), one major problem faced by semantic annotation of text is the fact that similar syntactic patterns might introduce different semantic interpretations and that similar meanings can be realized syntactically in many different ways.

3.4 Hybrid Approach for Automatic Relation Extraction

Confronting the challenges of extracting CDL.nl relations described above, in this Section, we present a hybrid approach: first, based on dependency analysis, a rule-based method is advanced for relation detection; secondly, we use a kernel-based classification method to assign a CDL.nl relation to each detected entity pair.

3.4.1 Rule-based Entity Pair Identification

Language processing has been going through syntactic processing, dependency analysis, and shallow semantic parsing. To find a relationship between entities in the level of semantic processing, we use dependency analysis as the basis to perform our relation detection task because it shows the head-modifier relations between words in the level of surface-syntactic processing in a word-to-word way.

In dependency parsing[77], the task is to create links between words and name the links according to their syntactic function. By identifying the syntactic head of each word in the sentence, the analysis result is represented in a dependency graph, where the nodes are the words of the input sentence and the arcs are the binary relations from the head to dependent. Often, but not always, it is assumed that all words except the root one have a syntactic head, which means that the graph will be a tree with the single independent word as the root. In labeled dependency parsing, a specific type (or

label) is assigned to each dependency relation that pertains between a head word and a dependent word.

Algorithm 1: $relationDetection$

Input: one sentence S

Output: entity pair list EP

parsing the sentence to get a dependency tree $DT =< V, E >$.

(V is a set of nodes and E is a set of directed edges.)

for *each non-terminal node* v_i *in* V **do**

  tag = headDetectionRules(v_i)

  (apply headDetection rules to decide if it is a headNode.)

  if *tag = 1* **then**

    tvs_i = tailDetectionRules(v_i)(get tailNode List headed by v_i)

    $EP \leftarrow [v_i, tv_i]$

return EP

Figure 3.4: Rule-based relation detection algorithm

Different from "all-pairs" and "stepwise" approaches, based on dependency tree structure generated from Connexor dependency parser[1], we present a rule-based method for relation detection that is done with a simple algorithm; it is depicted in Fig. 3.4.

We design two types of rules for detecting headNode and tailNode respectively. headDetectionRules is a set of rules we use to select nodes which have subtrees and omit those which cannot be headNodes by adopting a headNode stoplist. TailNodes are detected by using tailDetectionRules and a tailNode stoplist containing those which cannot be root nodes of subtrees of tail entities. We continue to check the immediate grandchildren until reaching the leaf nodes if the root node of a subtree is in the tail stoplist. Examples of headDetectionRules and tailDetectionRules are:

- Rule 1: node v_i is not in headNode stoplist; v_i has children nodes and at least one of its depend labels with children nodes is not in depend stoplist $\Rightarrow$ node is headNode; tag=1
- Rule 2: node v_j is not in tailNode stoplist and its depend label with v_i is not in depend stoplist $\Rightarrow$ put node v_j in tailNode list of v_i

[1] www.connexor.com

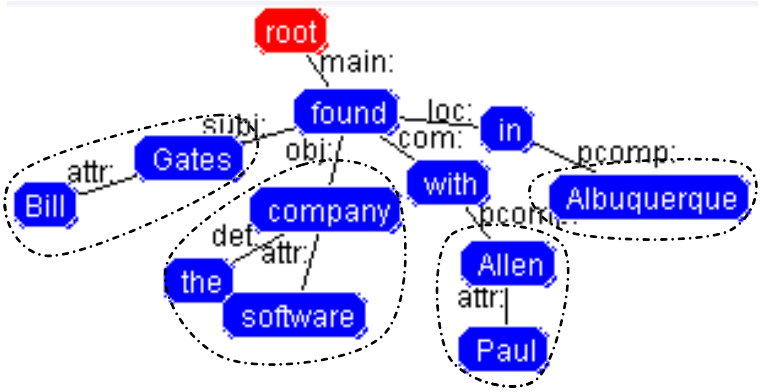

Figure 3.5: Relation detection example from Connexor parser

Finally, a simple post-processing is applied to correct the boundaries within which the dependency tree does not show correct relationships. As depicted in Fig. 3.5, for the sentence "Bill Gates found the software company with Paul Allen in Albuquerque", from the dependency tree, the following entity boundaries are generated from the dependency tree: [found, (Bill Gates)], [found, (the software company)], [found, (Paul Allen)], [found, Albuquerque] and [company, software].

3.4.2 Kernel Method for Relation Classification

In this subsection, facing the challenges of labeling each detected pair with a specific CDL.nl relation, we describe a relation classification approach which uses kernel functions to model diverse knowledge of three levels of language processing: syntactic analysis, dependency parsing and lexical construction.

3.4.2.1 Syntactic Kernel

As a benefit from the Connexor Parser, rich linguistic tags can be extracted as features to classify relations between entities. For each pair of entities of relation instances, we extract the following syntactic features and define a syntactic kernel to match two relation instances.

- Morphology Features

Morphological information tells the details of word forms used in text.

We use a vector to represent the morphology feature space: $X_{Morp} = (x_1, x_2, ..., x_{70})$. Where x_i corresponds to a tag and receives "0" or "1" value, and Connexor Parser defines 70 morphology tags. For each entity E, $X_{Morp}(E) = (x_{e1}, x_{e2}, ..., x_{e70})$. Where, $x_{ei} = 1$ if the set of morphology tags of the headword of E contains the tag of the *ith* position, all other tags not contained will be set to $x_{ei} = 0$.

- Syntax Features

Syntax describes both surface syntactic and syntactic function information of words. For example, *%NH* (nominal head) and *%>N* (determiner or premodifier of a nominal) are surface syntactic tags, *@SUB* (Subject) and *@F-SUBJ* (Formal subject) are syntactic function tags. The Connexor Parser defines 40 Syntax tags.

As dealing with morphology features, syntax features for an entity E are represented in a vector: $X_{Syn}(E) = (x'_{e1}, x'_{e2}, ..., x'_{e40})$. For two entities of a relation instance R, the syntactic feature vector $X(R)$ is defined as the concatenation of morphology and syntax vector:

$$X(R) = (X_{Morp}(E_1)X_{Morp}(E_2)X_{Syn}(E_1)X_{Syn}(E_2))$$

Then we define a syntactic kernel to match syntactic features between two relation instances R_1, R_2 by simply calculating the dot product of two vectors:

$$K_S(R_1, R_2) = X(R_1) \bullet X(R_2) \tag{3.1}$$

3.4.2.2 Dependency Kernel

A dependency relation specifies an asymmetric grammatical function relationship between a pair of words, where one word is a dependent of the other word, which is called its governor. We use tree structure to represent the dependency parse result also generated from Connexor Parser.

For each pair of entities of relation instances, to extract a dependency feature set F_D, we define a dependency token $DT = (dep, path)$, where dep contains two labels: one is the first depend label in the dependency path, which is governed directly by the headword of head entity; the other is the final label in the dependency path pointing to

#	Text	Baseform	Syntactic relation	Syntax and morphology
1	Bill	bill	attr:>2	@A> %>N N NOM SG
2	Gates	gates	subj:>3	@SUBJ %NH N NOM SG
3	found	find	main:>0	@+FMAINV %VA V PAST
4	the	the	det:>6	@DN> %>N DET
5	software	software	attr:>6	@A> %>N N NOM SG
6	company	company	obj:>3	@OBJ %NH N NOM
7	with	with	com:>3	@ADVL %EH PREP
8	Paul	paul	attr:>9	@A> %>N N NOM SG
9	Allen	allen	pcomp:>7	@<P %NH N NOM SG
10	in	in	loc:>3	@ADVL %EH PREP
11	Albuquerque	albuquerque	pcomp:>10	@<P %NH N NOM SG
12	<s>	<s>		

Figure 3.6: Syntactic analysis example

the headword of participant entity. Both are closest to representing the direct dependency functions of the entity pair. In addition, $path$ is the shortest path in the parse tree from the head entity to the other entity. We define a dependency kernel to match dependency features between two relation instances R_1, R_2 by matching the values:

$$K_D(R_1, R_2) = \sum_{i=1,2} I(DT_{1i}, DT_{2i}) \tag{3.2}$$

Where $I(x, y)$ is a binary string match operator that gives 1 if $x = y$ and 0 otherwise.

Fig. 3.6 portrays some examples of syntactic and dependency information of the sentence "*Bill Gates found the software company with Paul Allen in Albuquerque*". The 4th Column named syntactic relation of Fig. 3.6 shows dependency relations, all the dependency relations form in a dependency tree structure.

3.4.2.3 Lexical Kernel

To confront the problem that similar syntactic patterns might introduce different semantic interpretations and similar semantic interpretations might be represented in

different syntactic patterns, we use lexical meaning knowledge to address it in this section. Lexical meaning knowledge contains two kinds of information: word sense and semantic behavior[48].

We develop a lexical kernel to capture lexical meaning knowledge from two lexical resources - including WordNet, and UNLKB, built with extensive human effort over years of work. Each resource encodes a different kind of knowledge and presents its own advantages.

- WordNet

WordNet[31] is an on-line lexical system whose smallest unit is "synset", i.e. an equivalence class of word senses under the synonym relation. Synsets are organized by semantic relations such as Synonymy, Antonymy and Hyponymy. In WordNet 3.0, the total of all unique noun, verb, adjective, and adverb strings is actually 155287 along with 206941 word-sense pairs, containing 11529 verbs with 25047 verb-sense pairs. In this thesis, we use hypernymy and synonymy to represent word sense feature and also use synonymy to extend the later resource.

A vector W is defined to capture sense features containing word sense and hypernym senses of the headword of each entity: $W = (w_1, w_2..., w_n)$. Since each word might have many hypernym senses, in our experiments, we select the top four senses.

- UNLKB

Based on the CDL.nl semantic relation set, for each usage of the word, we define semantic behavior as a series of CDL.nl semantic relations in which the word participates. Because many words have different senses and usages they might have several semantic behaviors. The UNLKB[1] is a lexicon which organizes words in a hierarchical structure according to their semantic behaviors. It includes nouns, verbs, adjectives, and adverbs and associates semantic relations in behavior representation with word type restrictions. The total of all word-behavior pairs is about 65000, containing 15000 verb-behavior pairs. It implements the close relationship between syntax and semantics for nouns, verbs, adjectives, and adverbs explicitly. Here are some word-behavior pairs of word *give* in UNLKB:

[1] www.undl.org/unlsys/uw/unlkb.htm

give(agt>thing,obj>thing)
give(agt>thing,gol>person,obj>thing)
give(agt>thing,gol>thing,obj>thing)
give(agt>volitional thing,obj>action)

The word *give* has semantic behaviors of at least these four kinds. Furthermore, for the second behavior, it has *agent* relation with a thing-type word, *goal* relation with a person-type word and *object* relation with a thing-type word. Here, the type of a word is a hypernym word of the word.

Because UNLKB suffers from the coverage problem, we use the synonymy set from WordNet to extend them based on the assumption: words with identical senses tend to share the same behaviors.

A vector U is defined to capture the hierarchy hypermym of semantic behaviors of word: $U = (u_1, u_2..., u_n)$.

- Lexical Kernel Development

Both resources - WordNet and UNLKB - encode different kinds of knowledge. To explicitly capture these features, for the entity pair E_1, E_2, a new lexical feature vector $Y(R)$ is defined as the concatenation of both above lexical vectors:

$$Y(R) = (W(E_1)W(E_2)U(E_1)U(E_2))$$

Then we define the kernel to match lexical features between two relation instances R_1, R_2 by simply calculating the dot product of two vectors:

$$K_L(R_1, R_2) = Y(R_1) \bullet Y(R_2) \tag{3.3}$$

3.4.2.4 Composition Kernel

Having defined all the kernels representing syntactic, dependency and lexical processing results, we develop a composite kernel to combine and leverage individual kernels:

$$K = \alpha K_S + \beta K_D + \gamma K_L \tag{3.4}$$

This is the final kernel we used for this task. Trying with different α, β, γ values, we can observe performance of individual kernels and also of the composite kernel. Since all the individual kernels we defined can be seen directly/indirectly as matchings of features, it is clear that they are all valid kernels. And since the kernel function set is closed under linear combination, the composite kernels are also valid.

3.5 Experiments

3.5.1 Experimental Setting

Because this is the first work to extract CDL.nl relations from plain form text, currently no dataset exists for us to use for training and testing. After 46 person-days of discussion and manual annotation effort, we created a dataset[1] containing about 1700 sentences from Wikipedia documents. It was annotated with 13487 CDL.nl relations including 44 relation types. We evaluated the systems using ten-fold cross validation using this dataset.

To evaluate the performance of our relation classification method, we use one-vs.-all scheme in which each binary classifier will be trained for each relation label. The classifier evaluation is carried out using SVM-light software[46] with our syntactic, dependency, and lexical features.

3.5.2 Preliminary Experimental Results

The goals of our experiments are threefold: firstly, we intend to study the performance of a rule-based relation detection method. Secondly, we try to study how different kernels contribute to the classification task and we study how to leverage among syntactic, dependency and lexical features to get the best performance. Thirdly, the overall performance of relation extraction combining both steps is evaluated. For all of the purposes, three widely used evaluation measures (precision, recall and F-value) are computed.

[1] http://www.miv.t.u-tokyo.ac.jp/mem/yyan/CDLnl/

Table 3.2: Evaluation of rule-based relation detection

Task	Precision	Recall	F-value
Relation Detection	62.65	68.33	65.37

3.5.2.1 Evaluation of rule-based relation detection

For the first purpose of evaluation, the following quantities are considered to compute precision, recall, and F-value:

- p = the number of detected entity pairs.
- p+ = the number of detected entity pairs which are actual entity pairs.
- n = the number of actual entity pairs.

$$\text{Precision } (P) = \text{p+/p} \quad \text{Recall } (R) = \text{p+/n}$$
$$F\text{-value } (F) = 2 * P * R/(P + R)$$

The results of evaluating the test file are presented in Table 3.2. The performance is not high. Based on error analysis of the results, we conclude that the reasons might be the following. 1) Some special phrases must be treated as elemental entities, whereas our algorithm generates entity pairs inside of these phrases. 2) At the level of semantic information processing, we are trying to find deeper relationships than surface function relations. In some cases, when surface analysis is not able to reflect deep semantic information directly, we must improve our detection method. 3) Some of the detection errors resulted from failures by the dependency parser.

3.5.2.2 Evaluation of kernel-based relation classification

For the second purpose of evaluating the performance of kernel functions for relation classification, the process is divided into two steps: firstly, we intend to study the performance of individual kernels and watch if adding kernels continuously improves the performance. Secondly, we study how to leverage among syntactic, dependency and lexical features to get the best performance.

- **Individual Kernel Evaluation**

Table 3.3: Preliminary performance of individual kernels

Kernel	Precision	Recall	F-value
K_S	79.33	85.78	82.43
K_D	**83.62**	**83.56**	**83.59**
K_L	73.49	81.63	77.35
K_{S+D}	**85.63**	**85.91**	**85.77**
K_{S+D+L}	**86.35**	**87.43**	**86.89**

We test three individual kernels and the following two simple combination kernels:

$$K_{S+D} = K_S + K_D$$
$$K_{S+D+L} = K_S + K_D + K_L$$

The results are shown in Table 3.3 where we can get two observations, one is that using different feature set, the performance is different. This shows that each set contributes differently to our task. The performance of using dependency kernel is the best than using syntax kernel and lexical kernel. Another observation is that adding kernels continuously can improve the performance, which indicates they provide additional clues to the previous setup. While syntax kernel treats two entities as independent entities; the dependency kernel introduces dependency connection with grammatical function information between entities, so adding it to the syntax kernel boosts the performance. The lexical kernel introduces the meanings of entities, it helps in distinguishing semantic relations in case of same syntactic and dependency features using word sense and usage information, and so by adding it into the combination kernel, the performance is boosted.

- **Composition Kernel Evaluation**

Then for the composition kernel, we experiment several sets of α, β, γ values to compare the performance. As shown in Fig. 3.7, 10 times of evaluation on the testing set shows that our composition kernel yields different performance when α, β, γ are assigned with different values. The performance is the best in 7th time, with α, β, γ set up to 0.4, 0.6, 0.4. It also shows that after fine-tuning parameters, the performance (the

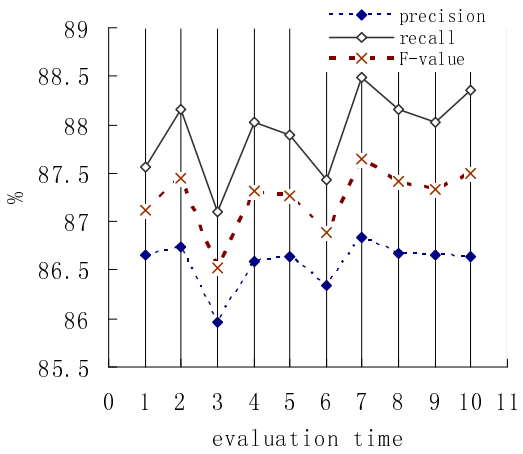

Figure 3.7: Performance with different values for α, β, γ

Table 3.4: Evaluation on incremental lexical features.

	Lexicon	Precision	Recall	F-value
A	No-Lexicon	85.63	85.91	**85.77**
B	A+WordNet	85.80	86.88	**86.34**
C	B+UNLKB	86.35	87.43	**86.89**

7th time evaluation) is better than that of using equal weights (the 6th time evaluation) for all kernels.

Although we do not experiment with all the values of each parameter, we can see that lexical features do not contribute much as expectation to the performance. The reason might be: the WordNet hierarchy is not a tree but rather includes multiple inheritances and a further complication is that several WordNet word-sense pairs or UNLKB word-behavior pairs are possible for a given head word. For example, "dog" has seven senses and the first sense [dog, domestic dog, Canis familiaris] has as hypernyms both [canine, canid] and [domestic animal, domesticated animal]. In our experiments, we simply use the first pair listed and the first hypernym listed. A word sense disambiguation module capable of distinguishing word senses and word behaviors might improve our results.

Table 3.5: Overall performance of relation extraction

TASK	Precision	Recall	F-value
Relation Detection (RD)	62.65	68.33	65.37
Relation Classification (RC)	86.35	87.43	86.89
RD + RC	51.62	57.94	54.60

In order to compare the contribution of each lexicon, we also evaluate each kind of lexical features. As shown in Table 3.4, the performance of using semantic behavior features from UNLKB lexicon is improved.

Through the preliminary experiments, we can see that despite confronting so many obstacles, CDL.nl relations were classified using kernel-based method with Precision, Recall, and F-values that are, respectively, 86.83%, 88.49%, 87.65%.

- **Overall performance of relation extraction**

For the third purpose of evaluation, Table 5.2 presents the preliminary result of the overall performance of our relation extraction approach by combining two steps. While the performance of the relation classification step is quite adequate, the performance of relation detection is low. Despite confronting so many obstacles, CDL.nl relations were extracted using our approach with Precision, Recall, and F-values that are, respectively, 51.62%, 57.94%, and 54.60%. Data analysis reveals that aside from dependency analysis, our method of relation detection can be improved by integrating diverse information from different levels of natural language processing.

We also show the performance of relation classification over some top relation types sorted by the number of instances. with the composition kernel, shown in Table 3.6. The classifier seems to perform better on the relation types with more annotated instances. Therefore, we hope to improve the classifier by bootstrap learning more instances for the relation types with rare instances.

3.6 Conclusions

In this thesis, to surmount the challenges of describing the concept structure of text into structured representation, we created a shallow semantic parser that (1) used a

Table 3.6: Relation Classification for Each Relation Type.

Relation	#ins	Pre	Rec	F-v	Relation	#ins	Pre	Rec	F-v
mod	3128	86.39	93.59	89.85	**obj**	2697	80.87	86.05	83.38
aoj	2069	83.72	70.94	76.80	**and**	1122	90.48	93.44	91.94
agt	1046	93.91	90.76	92.31	**man**	788	86.21	86.21	86.21
plc	446	91.89	87.18	89.47	**gol**	395	71.79	73.68	72.72
tim	321	87.10	77.14	81.81	**pur**	289	87.50	93.33	90.32
qua	269	78.95	83.33	81.08	**pos**	86	66.67	50.00	57.14
scn	71	100.0	85.71	92.30	**rsn**	65	57.14	80.00	66.64
src	63	87.50	87.50	87.50	**cnt**	61	100.0	28.57	44.44
dur	58	85.71	85.71	85.71	**bas**	49	50.00	50.00	50.00
met	47	60.00	100.0	75.00	**equ**	46	57.14	100.0	72.72
nam	41	57.14	100.0	72.72	**con**	41	100.0	100.0	100.0
tmt	25	83.33	100.0	90.91	**pof**	24	50.00	100.0	66.00
or	21	100.0	100.0	100.0	**All**	**13268**	**86.35**	**87.43**	**86.89**

new set of semantic relations of CDL.nl, which has better coverage than those of SRL, to represent the concept structure of text. In addition, (2) we proposed a hybrid relation extraction approach: a rule-based method is presented to detect all entity pairs between each of pair for which there exists a relationship; then, a kernel-based method is proposed to assign a CDL.nl relation to each detected entity pair. Experiments conducted using our manual dataset revealed that CDL.nl relations can be extracted with good performance by integrating diverse information from different levels of natural language processing.

Chapter 4

Unsupervised Relation Extraction by Mining Wikipedia Texts Using Information from the Web

Our approach in this thesis for natural language understanding is that: text in natural language form can be organized into syntactic and semantic structure, and then such well-organized form can be analyzed by software applications for various purposes. In the previous part, we have introduced a language to encode the knowledge or the relationships between the concepts implied in text to organize the text at a specific level of understanding. In this second part, we introduce an application of such intermediate representation to produce another structure of text at a higher level of understanding. Particularly, we introduce a system to extract facts from text, which are believed to be interested by human users. They can be used to answer factual questions such as: "Who is the founder of company XYZ?", "Show me the list of products of the company", and so on.

From this chapter, we present the second part of our work which is relation extraction from Wikipedia articles. We focus on extracting semantic relations between named entities such as "Chairman" relationship between "Bill Gates" and "Microsoft" which express the semantic information that "Bill Gates is the Chairman of Microsoft"; or "Spouse" relationship between "Bill Gates" and "Melinda Gates" which means that "Bill Gates married Melinda Gates".

4.1 Introduction

Machine learning approaches for relation extraction tasks require substantial human effort, particularly when applied to the broad range of documents, entities, and relations existing on the Web. Even with semi-supervised approaches, which use a large unlabeled corpus, manual construction of a small set of seeds known as true instances of the target entity or relation is susceptible to arbitrary human decisions. Consequently, a need exists for development of semantic information-retrieval algorithms that can operate in a manner that is as unsupervised as possible.

Currently, the leading methods in unsupervised information extraction collect redundancy information from a local corpus or use the Web as a corpus [4, 7, 24, 30, 66]. The standard process is to scan or search the corpus to collect co-occurrences of word pairs with strings between them, and then to calculate term co-occurrence or generate surface patterns. The method is used widely. However, even when patterns are generated from well-written texts, frequent pattern mining is non-trivial because the number of unique patterns is loose, but many patterns are non-discriminative and correlated. A salient challenge and research interest for frequent pattern mining is abstraction away from different surface realizations of semantic relations to discover discriminative patterns efficiently.

Linguistic analysis is another effective technology for semantic relation extraction, as described in many reports such as [47]; [11]; [41]; [61]. Currently, linguistic approaches for semantic relation extraction are mostly supervised, relying on prespecification of the desired relation or initial seed words or patterns from hand-coding. The common process is to generate linguistic features based on analysis of the syntactic features, dependency, or shallow semantic structure of text. Then the system is trained to identify entity pairs that assume a relation and to classify them into pre-defined relations. The advantage of these methods is that they use linguistic technologies to learn semantic information from different surface expressions.

As described herein, we consider integrating linguistic analysis with Web frequency information to improve the performance of unsupervised relation extraction. As [4] reported, “deep” linguistic technology presents problems when applied to heterogeneous text on the Web. Therefore, we do not parse information from the Web corpus, but from well written texts. Particularly, we specifically examine unsupervised

relation extraction from existing texts of Wikipedia[1] articles. Wikipedia resources of a fundamental type are of concepts (e.g., represented by Wikipedia articles as a special case) and their mutual relations. We propose our method, which groups concept pairs into several clusters based on the similarity of their contexts. Contexts are collected as patterns of two kinds: dependency patterns from dependency analysis of sentences in Wikipedia, and surface patterns generated from highly redundant information from the Web.

The main contributions of this part of work are as follows:

- Using characteristics of Wikipedia articles and the Web corpus respectively, our study yields an example of bridging the gap separating "deep" linguistic technology and redundant Web information for Information Extraction tasks.
- Our experimental results reveal that relations are extractable with good precision using linguistic patterns, whereas surface patterns from Web frequency information contribute greatly to the coverage of relation extraction.
- The combination of these patterns produces a clustering method to achieve high precision for different Information Extraction applications, especially for bootstrapping a high-recall semi-supervised relation extraction system.

The remainder of this chapter is organized as follows. Section 4.2 presents related work. Section 4.3 provides more precise definitions of the problem we intend to solve, through an analysis of the characteristics of Wikipedia articles. Section 4.4 presents an overview of our method and describes it in detail. In section 4.5, we report our exploratory experimental results. We conclude this chapter in section 4.6.

4.2 Related Work

[43] introduced a method for discovering a relation by clustering pairs of co-occurring entities represented as vectors of context features. They used a simple representation of contexts; the features were words in sentences between the entities of the candidate pairs.

[1] http://en.wikipedia.org/wiki/Main_Page

[79] presented an unsupervised algorithm for mining the Web for patterns expressing implicit semantic relations. Given a word pair, the output list of lexicon-syntactic patterns was ranked by pertinence, which showed how well each pattern expresses the relations between word pairs.

[23] proposed a method for unsupervised discovery of concept specific relations, requiring initial word seeds. That method used pattern clusters to define general relations, specific to a given concept. [24] presented an approach to discover and represent general relations present in an arbitrary corpus. That approach incorporated a fully unsupervised algorithm for pattern cluster discovery, which searches, clusters, and merges high-frequency patterns around randomly selected concepts.

The field of Unsupervised Relation Identification (URI)—the task of automatically discovering interesting relations between entities in large text corpora—was introduced by [43]. Relations are discovered by clustering pairs of co-occurring entities represented as vectors of context features. [68] showed that the clusters discovered by URI are useful for seeding a semi-supervised relation extraction system. To compare different clustering algorithms, feature extraction and selection method, [69] presented a URI system that used surface patterns of two kinds: patterns that test two entities together and patterns that test either of two entities.

In this chapter, we propose an unsupervised relation extraction method that combines patterns of two types: surface patterns and dependency patterns. Surface patterns are generated from the Web corpus to provide redundancy information for relation extraction. In addition, to obtain semantic information for concept pairs, we generate dependency patterns to abstract away from different surface realizations of semantic relations. Dependency patterns are expected to be more accurate and less spam-prone than surface patterns from the Web corpus. Surface patterns from redundancy Web information are expected to address the data sparseness problem. Wikipedia is currently widely used information extraction as a local corpus; the Web is used as a global corpus.

4.3 Characteristics of Wikipedia articles

Wikipedia is a multilingual, Web-based encyclopedia. It is written collaboratively by volunteers and is available for free under the terms of the GNU Free Documentation

License6. As of November 2009, the English Wikipedia contained more than 3 million articles. Each Wikipedia article is a single Web page and usually describes a single topic or entity.

Wikipedia, unlike the whole Web corpus, has several characteristics that markedly facilitate information extraction. First, as an earlier report [37] explained, Wikipedia articles are much cleaner than typical Web pages. Because the quality is not so different from standard written English, we can use "deep" linguistic technologies, such as syntactic or dependency parsing. Secondly, Wikipedia articles are heavily cross-linked, in a manner resembling cross-linking of the Web pages. [34] assumed that these links encode numerous interesting relations among concepts, and that they provide an important source of information in addition to the article texts.

To establish the background for this part of work, we start by defining the problem under consideration: relation extraction from Wikipedia. We use the encyclopedic nature of the corpus by specifically examining the relation extraction between the entitled concept (*ec*) and a related concept (*rc*), which are described in anchor text in this article. A common assumption is that, when investigating the semantics in articles such as those in Wikipedia (e.g. semantic Wikipedia [81]), key information related to a concept described on a page p lies within the set of links $l(p)$ on that page; particularly, it is likely that a salient semantic relation r exists between p and a related page $p' \in l(p)$.

Given the scenario we described along with earlier related works, the challenges we face are these: 1) enumerating all potential relation types of interest for extraction is highly problematic for corpora as large and varied as Wikipedia; 2) training data or seed data are difficult to label. Considering [24], which describes work to get the target word and relation cluster given a single ('hook') word, their method depends mainly on frequency information from the Web to obtain a target and clusters. Attempting to improve the performance, our solution for these challenges is to combine frequency information from the Web and the "high quality" characteristic of Wikipedia text.

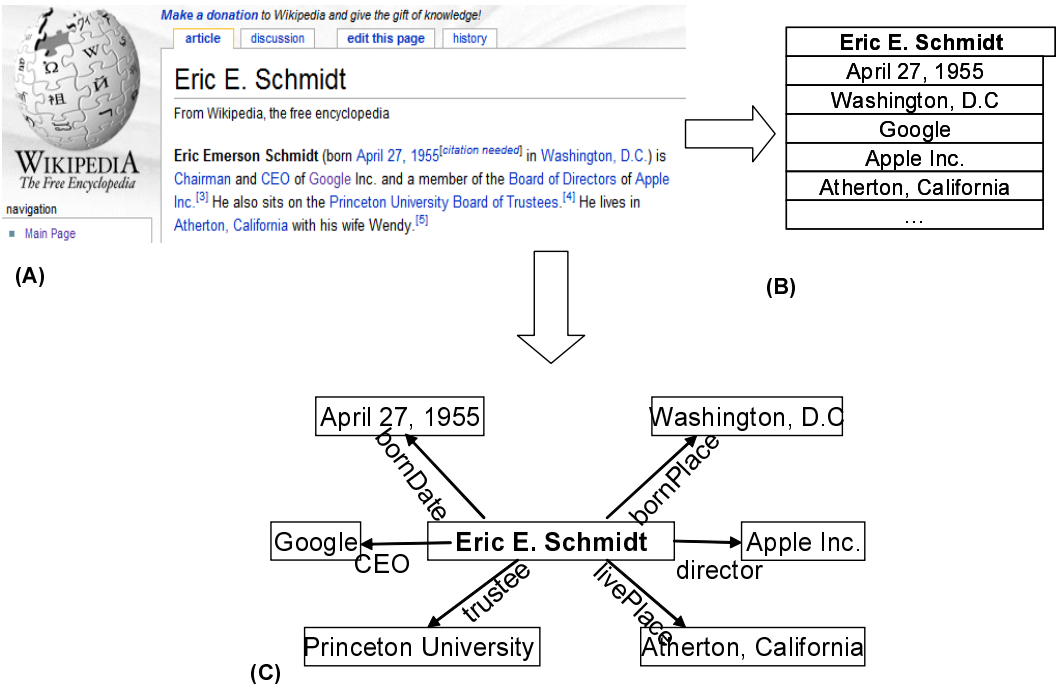

Figure 4.1: An example showing how we define the problem

4.4 Pattern Combination Method for Relation Extraction

With the scene and challenges stated, we propose a solution in the following way. The intuitive idea is that we integrate linguistic technologies on high-quality text in Wikipedia and Web mining technologies on a large-scale Web corpus. In this section, we first provide an overview of our method along with the function of the main modules. Subsequently, we explain each module in the method in detail.

4.4.1 Problem Definition

We formally define our task as open relation extraction from Wikipedia. Given a serials of Wikipedia documents in one domain, we aim to discover and enhance dominating binary relations shared in this domain with as less as possible human work. We define binary relation as a triple $< ec, rel, rc >$ in which ec is entitle concept described with

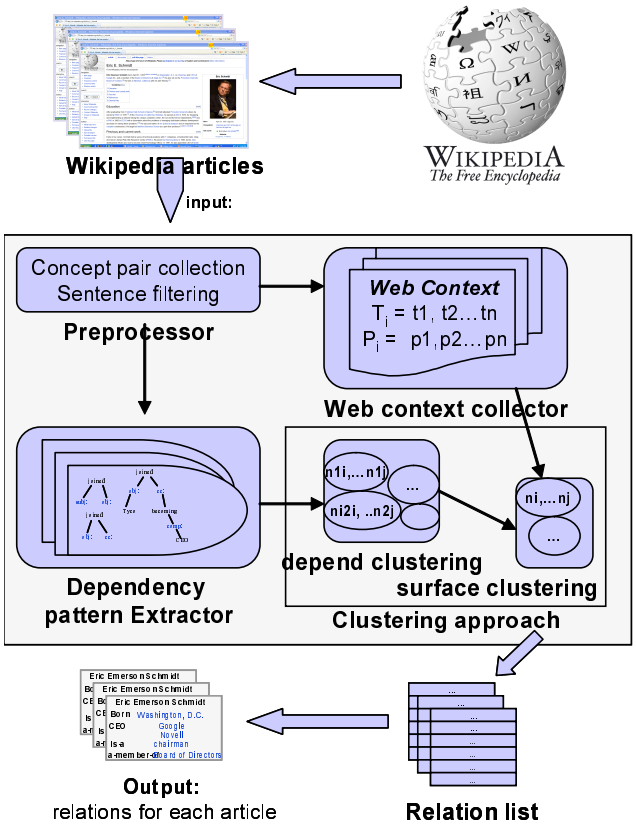

Figure 4.2: Framework of the proposed approach

the object Wikipedia article, and rc is a related concept and rel indicates the directed relationship between ec and rc. More specifically, As shown in the example in Fig. 4.1, for each particular entitled concept ("Eric E.Schmidt" in Fig. 4.1 (A)) in a Wikipedia article, the task is divided into two subtasks: (1) collecting related concepts (Fig. 4.1 (B)); (2) choose to label the relationships between some of them and the entitle concept with a specific relation type which maybe interesting to many people (Fig. 4.1 (C)).

4.4.2 Overview of the Method

Given a set of Wikipedia articles as input, our method outputs a list of concept pairs for each article with a relation label assigned to each concept pair. Briefly, the proposed approach has four main modules, as depicted in Fig. 4.2.

- **Text Preprocessor and Concept Pair Collector** preprocesses Wikipedia articles to split text and filter sentences. It outputs concept pairs, each of which has an accompanying sentence.
- **Web Context Collector** collects context information from the Web and generates ranked relational terms and surface patterns for each concept pair.
- **Dependency Pattern Extractor** generates dependency patterns for each concept pair from corresponding sentences in Wikipedia articles.
- **Clustering Algorithm** clusters concept pairs based on their context. It consists of the two sub-modules described below.
  - **Depend Clustering**, which merges concept pairs using dependency patterns alone, aiming at obtaining clusters of concept pairs with good precision;
  - **Surface Clustering**, which clusters concept pairs using surface patterns based on the resultant clusters of depend clustering. The aim is to merge more concept pairs into existing clusters with surface patterns to improve the coverage of clusters.

The key to our method lies in the complementarity of redundancy information from the Web and deep linguistic analysis for detecting and labeling relations in which a specified concept in Wikipedia participates.

4.4.3 Text Preprocessor and Concept Pair Collector

This module pre-processes Wikipedia article texts to collect concept pairs and corresponding sentences. Given a concept described in a Wikipedia article, our idea of preprocessing executes initial consideration of all anchor-text concepts linking to other Wikipedia articles in the article as related concepts that might share a semantic relation with the entitled concept. The link structure, more particularly, the structure of outgoing links, provides a simple mechanism for identifying relevant articles. We split text into sentences and select sentences containing one reference of an entitled concept and one of the linked texts for the dependency pattern extractor module.

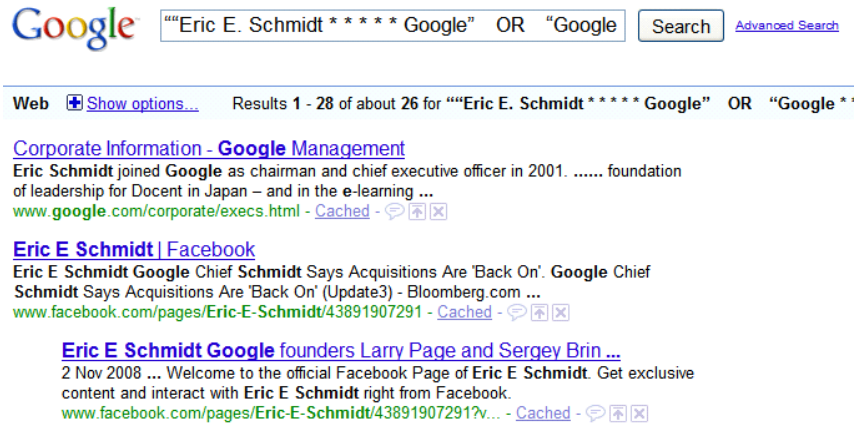

Figure 4.3: Snippets retrieved by querying with a sample entity pair'

4.4.4 Web Context Collector

Querying a concept pair using a search engine (Google), we characterize the semantic relation between the pair by leveraging the vast size of the Web. Our hypothesis is that there exist some key terms and patterns that provide clues to the relations between pairs. From the snippets retrieved by the search engine, we extract relational information of two kinds: ranked relational terms as keywords and surface patterns. Here surface patterns are generated with support of ranked relational terms. We take the entity pair "¡Eric E. Schmidt, Google¿" as an example. Fig. 4.3 shows the snippets retrieved by querying with ""Eric E. Schmidt ***** Google"OR"Google ***** Eric E. Schmidt"".

4.4.4.1 Relational Term Ranking

To collect relational terms as indicators for each concept pair, we look for verbs and nouns from qualified sentences in the snippets instead of simply finding verbs. Using only verbs as relational terms might engender the loss of various important relations, e.g. noun relations "CEO", "founder" between a person and a company. Therefore, for each concept pair, a list of relational terms is collected. Then all the collected terms of all concept pairs are combined and ranked using an entropy-based algorithm which is described in [14]. With their algorithm, the importance of terms can be assessed

using the entropy criterion, which is based on the assumption that a term is irrelevant if its presence obscures the separability of the dataset. After the ranking, we obtain a global ranked list of relational terms T_{all} for the whole dataset (all the concept pairs). For each concept pair, a local list of relational terms T_{cp} is sorted according to the terms' order in T_{all}. Then from the relational term list T_{cp}, a keyword t_{cp} is selected for each concept pair *cp* as the first term appearing in the term list T_{cp}. Keyword t_{cp} will be used to initialize the clustering algorithm in Section 4.4.6.1. For entity pair $< EricE.Schmidt, Google >$, the relational term list is "[CEO, Executive, Chief, Chairman, announce, lead, recruited, join, director, officer]".

4.4.4.2 Surface Pattern Generation

Because simply taking the entire string between two concept words captures an excess of extraneous and incoherent information, we use T_{cp} of each concept pair as a key for surface pattern generation. We classified words into Content Words (CWs) and Functional Words (FWs). From each snippet sentence, the entitled concept, related concept, or the keyword k_{cp} is considered to be a Content Word (CW). Our idea of obtaining FWs is to look for verbs, nouns, prepositions, and coordinating conjunctions that can help make explicit the hidden relations between the target nouns.

Surface patterns have the following general form.

$$[\textbf{CW1}]\ Infix_1\ [\textbf{CW2}]\ Infix_2\ [\textbf{CW3}] \tag{4.1}$$

Therein, $Infix_1$ and $Infix_2$ respectively contain only and any number of FWs. A pattern example is "*ec* assign *rc* as *ceo (keyword)*". All generated patterns are sorted by their frequency, and all occurrences of the entitled concept and related concept are replaced with "*ec*" and "*rc*", respectively for pattern matching of different concept pairs.

Table 4.1 presents examples of surface patterns for a sample concept pair. Pattern windows are bounded by CWs to obtain patterns more precisely because 1) if we use only the string between two concepts, it may not contain some important relational information, such as "ceo *ec* resign *rc*" in Table 4.1; 2) if we generate patterns by setting a windows surrounding two concepts, the number of unique patterns is often exponential.

Table 4.1: Surface patterns for a concept pair

Pattern	Pattern
ec ceo *rc*	*rc* found *ec*
ceo *rc* found *ec*	*rc* succeed as ceo of *ec*
rc be ceo of *ec*	*ec* ceo of *rc*
ec assign *rc* as ceo	*ec* found by ceo *rc*
ceo of *ec rc*	*ec* found in by *rc*

4.4.5 Dependency Pattern Extractor

In this section, we describe how to obtain dependency patterns for relation clustering. After preprocessing, selected sentences that contain at least one mention of an entitled concept or related concept are parsed into dependency structures. We define dependency patterns as sub-paths of the shortest dependency path between a concept pair for two reasons. One is that the shortest path dependency kernels outperform dependency tree kernels by offering a highly condensed representation of the information needed to assess their relation [11]. The other reason is that embedded structures of the linguistic representation are important for obtaining good coverage of the pattern acquisition, as explained in [20]; [89]. The process of inducing dependency patterns has two steps, as shown in Fig. 4.4

1. Shortest dependency path inducement. From the original dependency tree structure by parsing the selected sentence for each concept pair, we first induce the shortest dependency path with the entitled concept and related concept.

2. Dependency pattern generation. We use a frequent tree-mining algorithm [87] to generate sub-paths as dependency patterns from the shortest dependency path for relation clustering.

4.4.6 Clustering Algorithm for Relation Extraction

In this subsection, we present a clustering algorithm that merges concept pairs based on dependency patterns and surface patterns. The algorithm is based on k-means clustering for relation clustering.

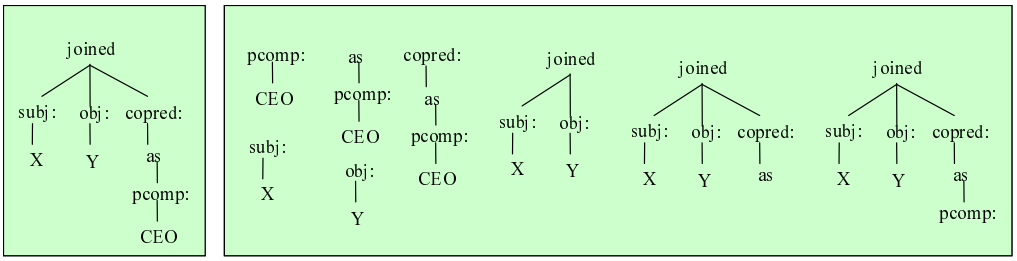

Figure 4.4: Dependency patterns for sample sentence "X joined Y as CEO."

The dependency pattern has the properties of being more accurate, but the Web context has the advantage of containing much more redundant information than Wikipedia. Our idea of concept pair clustering is a two-step clustering process: first it clusters concept pairs into clusters with good precision using dependency patterns; then it improves the coverage of the clusters using surface patterns.

4.4.6.1 Initial Centroid Selection and Distance Function Definition

The standard k-means algorithm is affected by the choice of seeds and the number of clusters k. However, as we claimed in the Introduction section, because we aim to extract relations from Wikipedia articles in an unsupervised manner, cluster number k is unknown and no good centroids can be predicted. As described in this work, we select centroids based on the keyword t_{cp} of each concept pair.

First of all, all concept pairs are grouped by their keywords t_{cp}. Let $G = \{G_1, G_2, ...G_n\}$ be the resultant groups, where each $G_i = \{cp_{i1}, cp_{i2}, ...\}$ identify a group of concept pairs sharing the same keyword t_{cp} (such as "CEO"). We rank all the groups by their number of concept pairs and then choose the top k groups. Then a centroid c_i is selected for each group G_i by Eq. 4.2.

$$c_i = \arg\max_{cp \in G_i} |\{cp_{ij} | (dis_1(cp_{ij}, cp) + \lambda * dis_2(cp_{ij}, cp)) <= D_z, 1 \leq j \leq |G_i|\}| \quad (4.2)$$

We assume a centroid for each group to be the concept pair which has the most other concept pairs in the same group that have distance less than D_z with it. Also, D_z

is a threshold to avoid noisy concept pairs: we assign it 1/3. To balance the contribution between dependency patterns and surface patterns, λ is used. The distance function to calculate the distance between dependency pattern sets DP_i, DP_j of two concept pairs cp_i and cp_j is dis_1. The distance is decided by the number of overlapped dependency patterns with Eq. 4.3.

$$dis_1(cp_i, cp_j) = 1 - \frac{|DP_i \cap DP_j|}{\sqrt{(|DP_i| * |DP_j|)}} \tag{4.3}$$

Actually, dis_2 is the distance function to calculate distance between two surface pattern sets of two concept pairs. To compute the distance over surface patterns, we implement the distance function $dis_2(cp_i, cp_j)$ in Fig. 4.5.

Algorithm 2: distance function $dis_2(cp_i, cp_j)$

Input: $SP_1 = \{sp_{11}, ..., sp_{1m}\}$(surface patterns of cp_i)
$SP_2 = \{sp_{21}, ..., sp_{2n}\}$ (surface patterns of cp_j)
Output: dis (distance between SP_1 and SP_2)
define a $m \times n$ distance matrix A: $\{A_{ij} = \frac{LD(sp_{1i}, sp_{2j})}{Max(|sp_{1i}|, |sp_{2j}|)}$, 1≤i≤m; 1≤j≤n$\}$;
$dis \leftarrow 0$
for $\min(m, n)$ *times* **do**
    (x, y) $\leftarrow \text{argmin}_{0<i<m;0<j<n} A_{ij}$;
    $dis \leftarrow dis$ + $A_{xy}/min(m, n)$;
    $A_{x*} \leftarrow 1$; $A_{*y} \leftarrow 1$;
return dis

Figure 4.5: Distance function over surface patterns

As shown in Fig. 4.5, the distance algorithm performs as: firstly it defines a $m \times n$ distance matrix A, then repeatedly selects two nearest sequences and sums up their distances. While computing dis_2, we use the Levenshtein distance LD to measure the difference of two surface patterns. The Levenshtein distance is a metric for measuring the amount of difference between two sequences (i.e., the so-called edit distance). Each generated surface pattern is a sequence of words. The distance of two surface patterns is defined as the fraction of the LD value to the length of the longer sequence.

For estimating the number of clusters k, we apply the stability-based criteria from [14] to decide the number of optimal clusters k automatically.

4.4.6.2 Concept Pair Clustering with Dependency Patterns

Given the initial seed concept pairs and cluster number k, this stage merges concept pairs over dependency patterns into k clusters. Each concept pair cp_i has a set of dependency patterns DP_i. We calculate distances between two pairs cp_i and cp_j using above the function $dis_1(cp_i, cp_j)$. The clustering algorithm is portrayed in Fig. 4.6. The process of depend clustering is to assign each concept pair to the cluster with the closest centroid and then recomputing each centroid based on the current members of its cluster. As shown in Figure 4.6, this is done iteratively by repeating both two steps until a stopping criterion is met. We apply the termination condition as: centroids do not change between iterations.

Algorithm 3: Depend Clustering

Input: $I = \{cp_1, ..., cp_n\}$(all concept pairs)
$C = \{c_1, ..., c_k\}$ (k initial centroids)
Output: $M_d : I \rightarrow C$ (cluster membership)
I_r (rest of concept pairs not clustered)
$C_d = \{c_1, ..., c_k\}$ (recomputed centroids)
while *stopping criterion has not been met* **do**
  for *each* $cp_i \in I$ **do**
    if $\min_{s \in 1..k} dis_1(cp_i, c_s) <= D_l$ **then**
      $M_d(cp_i) \leftarrow \text{argmin}_{s \in 1..k} dis_1(cp_i, c_s)$
    else
      $M_d(cp_i) \leftarrow 0$
  for *each* $j \in \{1..k\}$ **do**
    recompute c_j as the centroid of
      $\{cp_i | m_{loc}(cp_i) = j\}$
$I_r \leftarrow C_0$
return C and C_d

Figure 4.6: Clustering with dependency patterns

Because many concept pairs are scattered and do not belong to any of the top k clusters, we filter concept pairs with distance larger than D_l with the seed concept pairs. Such concept pairs are stored in C_0. We named the cluster of concept pairs Ir which are left to be clustered in the next step of clustering. After this step, concept

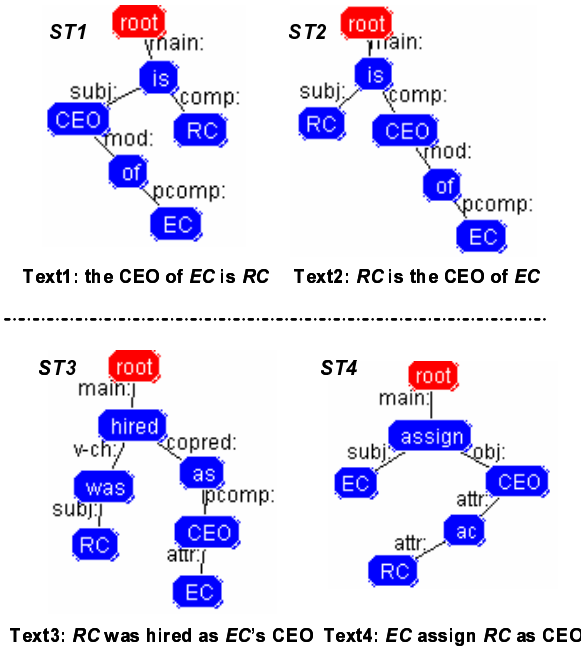

Figure 4.7: Example showing why surface clustering is needed

pairs with similar dependency patterns are merged into same clusters, see Fig. 4.7 (*ST1*, *ST2*).

4.4.6.3 Concept Pair Clustering with Surface Patterns

A salient difficulty posed by dependency pattern clustering is that concept pairs of the same semantic relation cannot be merged if they are expressed in different dependency structures. Figure 4.7 presents an example demonstrating why we perform surface pattern clustering. As depicted in Fig. 4.7, $ST1$, $ST2$, $ST3$, and $ST4$ are dependency structures for four concept pairs that should be classified as the same relation "CEO". However $ST3$ and $ST4$ can not be merged with $ST1$ and $ST2$ using the dependency patterns because their dependency structures are too diverse to share sufficient dependency patterns.

In this step, we use surface patterns to merge more concept pairs for each cluster to improve the coverage. Figure 4.8 portrays the algorithm. We assume that each concept pair has a set of surface patterns from the Web context collector module. As shown in

Figure 4.8, surface clustering is done iteratively by repeating two steps until a stopping criterion is met: using the distance function dis_2 explained in the preceding section, assign each concept pair to the cluster with the closest centroid and recomputing each centroid based on the current members of its cluster. We apply the same termination condition as depend clustering. Additionally, we filter concept pairs with distance greater than D_g with the centroid concept pairs.

Algorithm 4: Surface Clustering

Input: I_r (rest of concept pairs)
$C_d = \{c_1, ..., c_k\}$ (initial centroids)
Output: $M_s : I_r \rightarrow C$ (cluster membership)
$C_s = \{c_1, ..., c_k\}$ (final centroids)
while *stopping criterion has not been met* **do**
  for *each* $cp_i \in I_r$ **do**
    if $\min_{s \in 1..k} dis_2(cp_i, c_s) <= D_g$ **then**
      $M_s(cp_i) \leftarrow \text{argmin}_{s \in 1..k} dis_2(cp_i, c_s)$
    else
      $M_s(cp_i) \leftarrow 0$
  for *each* $j \in 1..k$ **do**
    recompute c_j as the centroid of cluster
      $\{cp_i | M_d(cp_i) = j \vee M_s(cp_i) = j\}$
return clusters C

Figure 4.8: Clustering with surface patterns

Finally we have k clusters of concept pairs, each of which has a centroid concept pair. To attach a single relation label to each cluster, we use the centroid concept pair.

4.5 Experiments

We apply our algorithm to two categories in Wikipedia: “American chief executives” and “Companies”. Both categories are well defined and closed. We conduct experiments for extracting various relations and for measuring the quality of these relations in terms of precision and coverage. We use coverage as an evaluation instead of using recall as a measure. The coverage is used to evaluate all correctly extracted concept

pairs. It is defined as the fraction of all the correctly extracted concept pairs to the whole set of concept pairs. To balance between precision and coverage of clustering, we integrate two parameters: D_l, D_g.

We downloaded the Wikipedia dump as of December 3, 2008. The performance of the proposed method is evaluated using different pattern types: dependency patterns, surface patterns, and their combination. We compare our method with [69]'s URI method. Their algorithm outperformed that presented in the earlier work using surface features of two kinds for unsupervised relation extraction: features that test two entities together and features that test only one entity each. For comparison, we use a k-means clustering algorithm using the same cluster number k.

4.5.1 Wikipedia Category: "American chief executives"

We choose appropriate D_l(concept pair filter in depend clustering) and D_g(concept pair filter in surface clustering) in a development set. To balance precision and coverage, we set 1/3 for both D_l and D_g.

The 526 articles in this category are used for evaluation. We obtain 7310 concept pairs from the articles as our dataset. The top 18 groups are chosen to obtain the centroid concept pairs. Of these, 15 binary relations are the clearly identifiable relations shown in Table 4.2, where # Ins. represents the number of concept pairs clustered using each method, and *pre* denotes the precision of each cluster.

The proposed approach shows higher precision and better coverage than URI in Table 4.2. This result demonstrates that adding dependency patterns from linguistic analysis contributes more to the precision and coverage of the clustering task than the sole use of surface patterns.

To examine the contribution of dependency patterns, we compare results obtained with patterns of different kinds. Table 4.3 shows the precision and coverage scores. The best precision is achieved by dependency patterns. The precision is markedly better than that of surface patterns. However, the coverage is worse than that by surface patterns. As we reported, many concept pairs are scattered and do not belong to any of the top k clusters, the coverage is low.

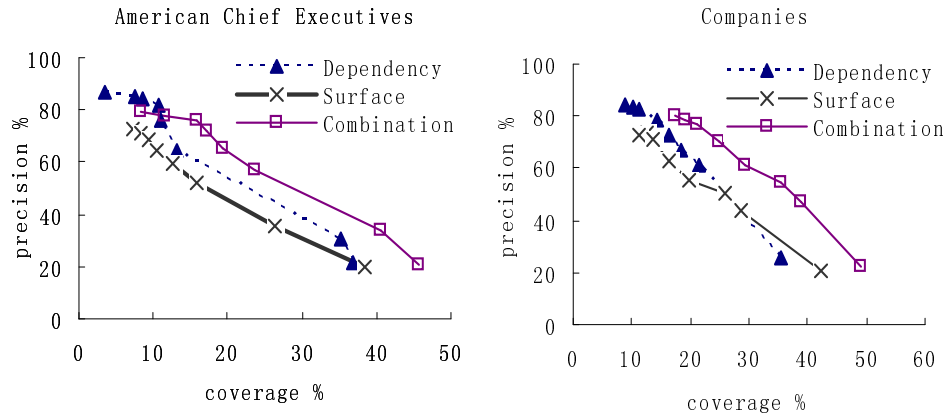

Figure 4.9: Precision-coverage curves on two categories

4.5.2 Wikipedia Category: "Companies"

We also evaluate the performance for the "Companies" category. Instead of using all the articles, we randomly select 434 articles for evaluation and 4073 concept pairs from the articles form our dataset for this category. We also set D_l and D_g to 1/3. Then 28 groups are chosen. For each group, a centroid concept pair is obtained. Finally, of 28 clusters, 25 binary relations are clearly identifiable relations. Table 4.4 presents some relations.

Our clustering algorithms use two filters D_l and D_g to filter scattering concept pairs. In Table 4.4, we present that concept pairs are clustered with good precision. As in the first experiments, the combination of dependency patterns and surface patterns contribute greatly to the precision and coverage. Table 4.5 shows that, using dependency patterns, the precision is the highest (82.58%), although the coverage is the lowest.

We then consider how performance changes for different values of D_l and D_g. The results are given as graphs in Fig. 4.9. It show the precision and coverage over different pattern sets for each categories. The performance is boosted with the combination of patterns in precision and coverage.

Fig. 4.10 illustrates an example showing the final output of our system for each en-

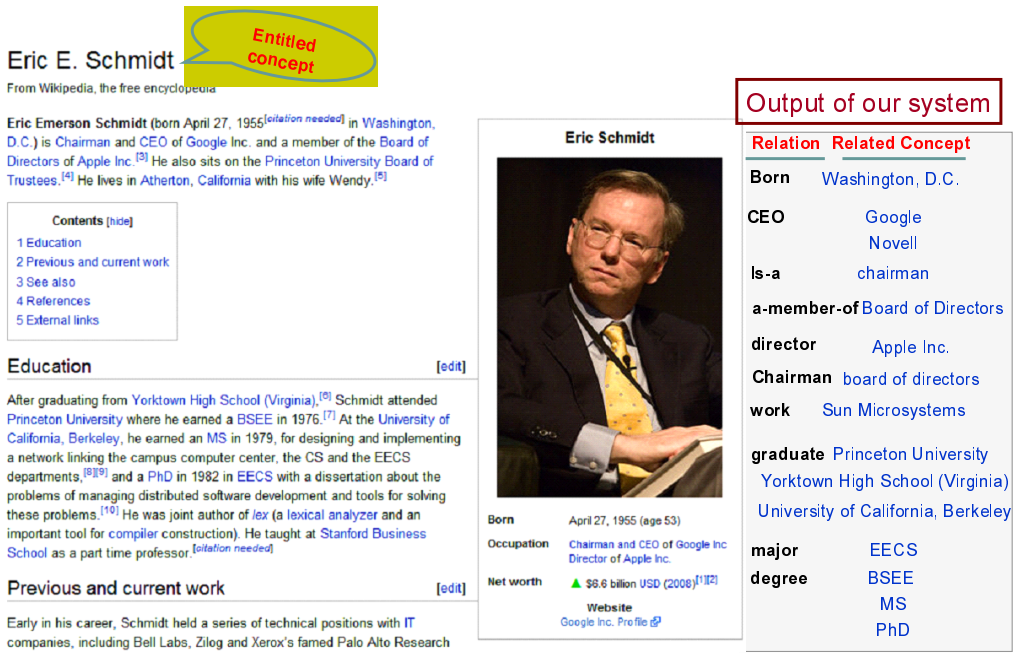

Figure 4.10: Final Output Example for One Concept "Eric E. Schmidt"

titled concept "Eric E. Schmidt". Comparing with the relations from infobox section, our system extract more interesting relations such as he graduated from Princeton University and he holds a PhD degree. However since only hypertexts are considered as related concept, we don't consider the relation between entity pair "¡Eric E. Schmidt, April 27, 1955¿".

All experimental results support our idea mainly in two aspects: 1) Dependency analysis can abstract away from different surface realizations of text. In addition, embedded structures of the dependency representation are important for obtaining a good coverage of the pattern acquisition. Furthermore, the precision is better than that of the string surface patterns from Web pages of various kinds. 2) Surface patterns are used to merge concept pairs with relations represented in different dependency structures with redundancy information from the vast size of Web pages. Using surface patterns, more concept pairs are clustered, and the coverage is improved.

4.6 Conclusions

To discover a range of semantic relations from a large corpus, we present an unsupervised relation extraction method using deep linguistic information to alleviate surface and noisy surface patterns generated from a large corpus, and use Web frequency information to ease the sparseness of linguistic information. We specifically examine texts from Wikipedia articles. Relations are gathered in an unsupervised way over patterns of two types: dependency patterns by parsing sentences in Wikipedia articles using a linguistic parser, and surface patterns from redundancy information from the Web corpus using a search engine. We report our experimental results in comparison to those of previous works. The results show that the best performance arises from a combination of dependency patterns and surface patterns. The combination of these patterns allows the clustering method to achieve high precision for bootstrapping a high-recall semi-supervised relation extraction system.

Table 4.2: Results for the category: “American chief executives”

method	Existing method (Rosenfeld et al.)		Proposed method (Our method)	
Relation (sample)	# Ins.	pre	# Ins.	pre
chairman (*x be chairman of y*)	434	63.52	547	68.37
ceo (*x be ceo of y*)	396	73.74	423	77.54
bear (*x be bear in y*)	138	83.33	276	86.96
attend (*x attend y*)	225	67.11	313	70.28
member (*x be member of y*)	14	85.71	175	91.43
receive (*x receive y*)	97	67.97	117	73.53
graduate (*x graduate from y*)	18	83.33	92	88.04
degree (*x obtain y degree*)	5	80.00	78	82.05
marry (*x marry y*)	55	41.67	74	61.25
earn (*x earn y*)	23	86.96	51	88.24
award (*x won y award*)	23	43.47	46	84.78
hold (*x hold y degree*)	5	80.00	37	72.97
become (*x become y*)	35	74.29	37	81.08
director (*x be director of y*)	24	67.35	29	79.31
die (*x die in y*)	18	77.78	19	84.21
all	1510	68.27	2314	75.63

Table 4.3: Performance of different pattern types

Pattern type	#Instance	Precision	Coverage
dependency	1127	84.29	13.00%
surface	1510	68.27	14.10%
Combined	2314	75.63	23.94%

Table 4.4: Results for the category: "Companies"

Method	Existing method (Rosenfeld et al.)		Proposed method (Our method)	
Relation (sample)	# Ins.	pre	# Ins.	pre
found (*found x in y*)	82	75.61	163	84.05
base (*x be base in y*)	82	76.83	122	82.79
headquarter (*x be headquarter in y*)	23	86.97	120	89.34
service (*x offer y service*)	37	51.35	108	69.44
store (*x open store in y*)	113	77.88	88	72.72
acquire (*x acquire y*)	59	62.71	70	64.28
list (*x list on y*)	51	64.71	67	70.15
product (*x produce y*)	25	76.00	57	77.19
CEO (*ceo x found y*)	37	64.86	39	66.67
buy (*x buy y*)	53	62.26	37	56.76
establish (*x be establish in y*)	35	82.86	26	80.77
locate (*x be locate in y*)	14	50.00	24	75.00
all	685	71.03	1039	76.87

Table 4.5: Performance of different pattern types

Pattern type	#Instance	Precision	Coverage
dependency	551	82.58	11.17%
surface	685	71.03	11.95%
Combined	1039	76.87	19.61%

Chapter 5

Multi-View Clustering with Web and Linguistic Features for Relation Extraction

There are many multi-view learning method that are used in information extraction. In this chapter, we study another unsupervised method by integrating frequency information from Web with linguistic analysis on Wikipedia texts in a multi-view co-clustering way for our open relation extraction from Wikipedia. One clustering is feature clustering by automatically learning clustering functions for Web features, linguistic features simultaneously. The other clustering is relation clustering, using the feature clustering functions to define learning function for relation extraction.

5.1 Introduction

Recent attention to automatically harvesting semantic resources has encouraged Data Mining and Natural Language Processing researchers to develop algorithms for it. Many efforts have also focused on extracting semantic relations between entities, such as *birth_date* relation, *CEO* relation, and other relations. Semantic relation extraction is also becoming an important component in various applications of Web mining [62] and NLP.

Currently one type of the leading methods in relation extraction are based on collecting redundancy information from a local corpus or use the Web as corpus [66];

[4]; [7]; [24]. Let us call them Web mining-based methods. The standard process is to scan or search the corpus to collect co-occurrences of word pairs with strings between them, then calculate term co-occurrence or generate textual patterns. In order to clearly distinguish from linguistic features below, let us call them Web features. For example, given an entity pair x, y with $Spouse$ relation, string "*x is married to y*" is a Web feature example. The method is used widely, however, even when patterns are generated from good-written texts, frequent pattern mining is non-trivial since the number of unique patterns is loose but many are non-discriminative and correlated. One of the main challenges and research interest for frequent pattern mining is how to abstract away from different surface realizations of semantic relations to discover discriminative patterns efficiently.

Another type of leading methods are using linguistic analysis for semantic relation extraction (see e.g., [47]; [11]; [41]; [61]). Let us call them linguistic-based methods. Currently, linguistic-based methods for semantic relation extraction are almost all supervised or semi-supervised, relying on pre-specification of the desired relationship or hand-coding initial seed words or features. The main process is to generate linguistic features based on the analysis of the syntactic, dependency or shallow semantic structure of text, then through training to identify entity pairs which assume a relationship and classify them into pre-defined relationships. For example, given an entity pair x, y and the sentence "*x is the wife of y*", syntactic, dependency features will be generated by analysis of the sentence. The advantage of these methods is using linguistic technologies to learn semantic information from different surface expressions.

Different from these relation extraction methods, in this chapter, we address a novel view of relation extraction task, where we take linguistic features and Web features of entity pairs as two separate views to enhance the clustering performance of extracting relations. In our problem, we do not have any labeled data or pairwise supervisory constraint knowledge. From Web view, a clustering operation on the target data can be performed using Web-based methods; on the other hand, from linguistic view, a clustering operation on the target data can be performed using linguistic-based method. The challenge is how to make use of both views to improve the performance.

Our solution for this two-view clustering problem is to perform two learning tasks through co-clustering. One is to merge features into clusters by perform co-clustering between Web features and linguistic features. The other is to cluster entity pairs by

co-clustering between entity pairs and feature (Web&linguistic) spaces. We extend two co-clustering algorithms for our solution. One is the information theoretic co-clustering algorithm [27] which minimizes loss in mutual information before and after clustering. The other is self-taught clustering algorithm [22] which performs clustering on a set of target data with auxiliary data simultaneously to allow the feature representation from the auxiliary data to influence the target data through a common set of features. Separate from those two works, we introduce a multi-view co-clustering approach which consists of two steps, we call it dual co-clustering. In the first step it automatically learns clustering functions for Web features, linguistic features and entity pairs simultaneously. Then in the second step, the feature clustering functions are used to learn a relation clustering as the final objective function. Our experiments on a dataset from Wikipedia corpus demonstrate the superiority of our clustering approach comparing with several state-of-the-art clustering methods.

The main contributions of this part of work are as follows:

- We propose a multi-view co-clustering algorithm. One is learning clustering functions for Web features and linguistic features simultaneously. The other is learning a clustering function for entity pairs based on feature clustering functions.

- Based on these algorithms, we construct an integrated framework for relation extraction task combining with Web features and linguistic features. The whole workflow is an instance of multi-view unsupervised learning. To the best of our knowledge, our approach is novel for various machine learning applications, especially for semantic relation extraction task.

- Our study suggests another example to bridge the gap between Web mining technology and "deep" linguistic technology for information extraction tasks. It shows how deep linguistic features can be combined with features from the whole Web corpus to improve the performance of information extraction tasks.

The remainder of the chapter is organized as follows. In section 5.2 we will consider related work of this work. In section 5.3 we define the problem formulation and present out our solution. In section 5.4 we will report on our experimental results. Finally, in section 5.5 we will conclude this work.

5.2 Related Work

In this section, we review several past research works that are related to our work, including, Web-based clustering, linguistic-based clustering and multi-view clustering.

The field of Unsupervised Relation Identification (URI) - the task of automatically discovering interesting relations between entities in a large text corpora was introduced by [43]. In [68] they showed that the clusters discovered by URI can be used for seeding a semi-supervised relation extraction system. To compare different clustering algorithm, feature extraction and selection method, the authors in [69] presented a URI system which used two kinds of surface patterns: patterns that test two entities together and patterns that test only one entity each. [23] proposed a method for unsupervised discovery of concept specific relations, requiring initial word seeds. They used pattern clusters to define general relationships, specific to a given concept. [24] presented an approach to discover and represent general relationships present in an arbitrary corpus. They presented a fully unsupervised algorithm for pattern cluster discovery, which searches, clusters and merges high frequency words-based patterns around randomly selected concepts.

Although linguistic-based relation extraction approaches for semantic relation extraction are almost supervised or semi-supervised, [15] presented an application of spectral clustering technique to unsupervised relation extraction problem, making use of various lexical and syntactic features from the contexts. [71] used simple predicate-argument patterns around the entities of candidate pairs. Their system worked on news articles, and improves its accuracy by looking at multiple news sources describing the same event. [59] built lexically-specific features by looking for verbs, prepositions, and coordinating conjunctions that can help make explicit the hidden relations between the target nouns.

Another related field is multi-view clustering[26]; [27]; [25]; [50]. Multiple view unsupervised learning is a fairly new topic. There is several work on multiple view clustering. Co-clustering techniques, which aim to cluster different types of data simultaneously by making efficient use of the relationship information, are proposed. [26] proposed a Bipartite Spectral Graph Partitioning approach to co-cluster words and documents. [27] presented the information theoretic co-clustering algorithm. With their information theoretic co-clustering, the objective function of co-clustering is defined

as minimizing loss in mutual information between entity pairs and features, before and after co-clustering. [25] also assumes two independent views for a multiple view data set and proposes a spectral clustering algorithm which creates a bipartite graph and is based on the minimizing-disagreement idea.

In this study, we propose a multi-view co-clustering approach for relation extraction task based on a combination of two types of features. On the one hand, Web features are generated from the Web information to provide frequency information. On the other hand, linguistic features are generated from local sentences by linguistic analysis to abstract information away from surface realizations of texts.

5.3 Dual Co-clustering Approach for Multi-view Learning

In this section, we present a dual co-clustering approach for relation extraction task based on two kinds of generated features: Web features and linguistic features.

5.3.1 Problem Formulation and Outline of the Proposed Approach

We define the multi-view relation clustering task. The task is that given a target dataset of entity pairs such as "Bill Gates & Microsoft", first we generate Web features and linguistic features from contexts of each entity pair, then cluster all the entity pairs into groups based on these features, each group represents a relationship, such as "CEO". Let X_{all} be a discrete random variable, taking values from the target data set $\{x_1, ...x_l\}$ which contains all entity pairs to be labeled with their relation types. We are interested in clustering X_{all} into L clusters, each of which represents one relation type. Let Y and Z be two discrete random variables, taking values from two value set $\{y_1, ...y_m\}$ and $\{z_1, ...z_n\}$, that respectively corresponds to two different feature spaces of X_{all}. Y represent features from Web frequency information, Z represent features from linguistic analysis. Respectively with only Web features or with only linguistic features, X_{all} will be clustered into L clusters in two different ways. However, Web and linguistic features usually represent two aspects of the meaningful of the same relations, thus there must be some deep connection between them. The main task of this work is that

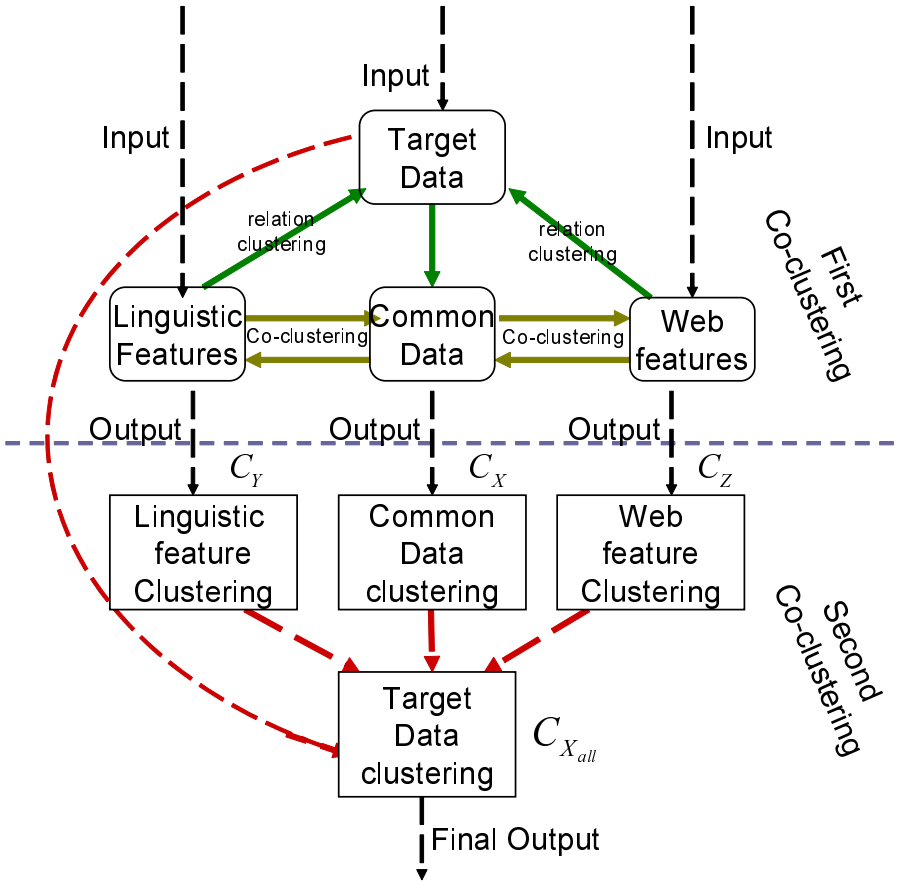

Figure 5.1: Outline of the proposed multi-view co-clustering approach.

given X_{all} with its feature spaces Y and Z, how to learn the connection between Web features and linguistic features to help perform clustering for relation extraction.

Our novel idea is to use another variable X, which works as an intermediate variable, to explore the deep connection between Web and linguistic features. It is used for information transformation among Web features, linguistic features and the target data set. Let X be a discrete random variable, taking values from value sets $\{x_1, ..., x_p\}$, which we call common data, corresponding to the shared entity pairs after perform relation clustering over Web features and linguistic features separately. The common data is a subset of the target data. Section 5.3.2 will explain how to obtain the common data in detail.

Fig. **??** shows the outline of our solution. The proposed approach consists of

two co-clustering steps: co-clustering learning for feature clusterings and co-clustering learning for relation clustering. In the first step, we are interested in simultaneously clustering X into L clusters, Y into (at most) M clusters, and Z into (at most) N clusters. In other words, we are interested in finding clustering functions C_X, C_Y and C_Z. The second step is to reach our objective which is to find a good clustering function $C_{X_{all}}$ for the whole target data, with the support of clustering functions C_X, C_Y and C_Z from the previous step. For brevity, in the following, we will use $\tilde{X_{all}}$, $\tilde{X}$, $\tilde{Y}$ and $\tilde{Z}$ to denote $C_{X_{all}}(X_{all})$, $C_X(X)$, $C_Y(Y)$ and $C_Z(Z)$, respectively. In other words, we are interested in firstly finding maps C_X, C_Y and C_Z and then finding map $C_{X_{all}}$:

$$C_X : \{x_1, ..., x_p\} \rightarrow \{\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_L\} \tag{5.1}$$

$$C_Y : \{y_1, ..., y_m\} \rightarrow \{\tilde{y}_1, \tilde{y}_2, ...\tilde{y}_M\} \tag{5.2}$$

$$C_Z : \{z_1, ..., z_n\} \rightarrow \{\tilde{z}_1, \tilde{z}_2, ...\tilde{z}_N\} \tag{5.3}$$

$$C_{X_{all}} : \{x_1, ..., x_l\} \rightarrow \{\tilde{x}_{a1}, \tilde{x}_{a2}, ...\tilde{x}_{aL}\} \tag{5.4}$$

5.3.2 Initialization of Common Data

Algorithm 5: The Common Data Initialization Algorithm

Input: $\tilde{X_1} = \{\tilde{x}_{11}, \tilde{x}_{12}, ..., \tilde{x}_{1L}\}$(clustering target data based on only Web features)

$\tilde{X_2} = \{\tilde{x}_{21}, \tilde{x}_{22}, ..., \tilde{x}_{2L}\}$(clustering target data based on only linguistic features)

Output: common data clustering $\tilde{X}$

define a $L \times L$ similarity matrix A: $\{A_{ij} = |(\tilde{x_{1i}} \bigcap \tilde{x_{2j}})|$ 1≤i≤L; 1≤j≤L$\}$;

$\tilde{X} = \phi$

for L *times* **do**

  (a, b) = $\text{argmax}_{0<i,j<L} A_{ij}$;

  $\tilde{X}$ = $\tilde{X}$ + $(\tilde{x_{1a}} \bigcap \tilde{x_{2b}})$;

  A_{a*} = 0; A_{*b} = 0;

return $\tilde{X}$

Figure 5.2: Common data initialization

The common data set is important for information connection between Web feature space and linguistic feature space. We initialize common data X and clustering

function C_X on X by three steps:

- Step 1: perform clustering operations on the target data over Web features and linguistic features separately;

$$C_{XY} : \{x_1, ..., x_l\} \rightarrow \{\tilde{x}_{11}, \tilde{x}_{12}, ...\tilde{x}_{1L}\}$$
$$C_{XZ} : \{x_1, ..., x_l\} \rightarrow \{\tilde{x}_{21}, \tilde{x}_{22}, ...\tilde{x}_{2L}\}$$

- Step 2: take two above clustering results as input for the common data initialization algorithm in Figure 5.2, we get the output which is a set of relation clusters.

  Algorithm 5 details the process involved in this initialization. The input is two sets of relation clusters $\tilde{X_1}$ and $\tilde{X_2}$ resulting from Step 1. The algorithm starts with defining a similarity matrix by counting the shared number of entity pairs between each pair of clusters from $\tilde{X_1}$ and $\tilde{X_2}$. The main loop then starts at line 3 and iterates L times. In each iteration, the entry A_{ab} with the largest value is chosen. The common entity pairs of ath cluster from $\tilde{X_1}$ and $\tilde{X_2}$ will form a new relation cluster, and then be added into the common cluster set $\tilde{X}$.

$$C_{XY} \wedge C_{XZ} : \{x_1, ..., x_l\} \rightarrow \{\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_L\} \tag{5.5}$$

- Step 3: simply from Equation 5.5, release all the entity pairs from the common cluster set to collect the common data in Equation 5.6. The initial clustering function for the common data is formulated in Equation 5.7.

$$\{\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_L\} \Rightarrow \{x_{c1}, ..., x_{cp}\} \tag{5.6}$$
$$C_X^0 : \{x_{c1}, ..., x_{cp}\} \rightarrow \{\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_L\} \tag{5.7}$$

5.3.3 Objective Function for Clustering Algorithm

We extend the information theoretic co-clustering [27] and self-taught clustering [22] to model our dual co-clustering learning algorithm. In the information theoretic co-clustering, the objective function of co-clustering is defined as minimizing loss in mutual information between entity pairs and features, before and after co-clustering. Formally, using the target data X and their feature space Y for illustration, the objective function can be expressed as:

$$I(X, Y) - I(\tilde{X}, \tilde{Y}) \tag{5.8}$$

where $I(.,.)$ denotes the mutual information between two random variables [18] that $I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$. Moreover, $I(\tilde{X}, \tilde{Y})$ corresponds to the joint probability distribution $p(\tilde{X}, \tilde{Y})$ which is defined as:

$$p(\tilde{x}, \tilde{y}) = \sum_{x \in \tilde{x}} \sum_{y \in \tilde{y}} p(x,y) \tag{5.9}$$

[22] extended the information theoretic co-clustering [27] to model a self-taught clustering algorithm. They model their self-taught clustering algorithm as performing co-clustering operations on the target data X and auxiliary data Y, simultaneously, while the two co-clusters share the same features clustering $\tilde{Z}$ on the feature set Z. Their objective function is formulated as:

$$I(X,Z) - I(\tilde{X}, \tilde{Z}) + \lambda[I(Y,Z) - I(\tilde{Y}, \tilde{Z})] \tag{5.10}$$

λ is a trade-off parameter to balance the influence between the target data and the auxiliary data. Z is used as the bridge to connect the knowledge between the target and auxiliary data.

In this work, we model our multi-view co-clustering learning algorithm in a two-step of clustering process: feature clustering and relation clustering.

- Learning Feature Clustering Functions

In the first step, we model our feature clustering as performing co-clustering operations on the common data X, feature set Y and feature set Z, simultaneously, while the two clusterings on Y and Z share a common relation clustering $\tilde{X}$ on the target data. The objective function for feature clustering defined as minimizing loss in mutual information between entity pairs and features can be formulated as:

$$I(X,Y) - I(\tilde{X}, \tilde{Y}) + \lambda[I(X,Z) - I(\tilde{X}, \tilde{Z})] \tag{5.11}$$

In Equation 5.11, $I(X,Y) - I(\tilde{X}, \tilde{Y})$ is computed on the clustering over only Web feature space Y on the common data, while $I(X,Z) - I(\tilde{X}, \tilde{Z})$ is computed over only linguistic feature space Z. We also use λ as a trade-off parameter to balance the contribution between Web features and linguistic features, which we will test in our experiments. The objective is to find maps C_Y and C_Z towards a common relation

clustering C_X. Intuitively, in an ideal way, targeting on the common data, the clustering function C_Y over Web features and C_Z over linguistic features will lead to the same clustering result. This restriction enables us to build a "bridge" to connect the knowledge between two feature spaces.

Our remaining task is to minimize the value of the objective function in Equation 5.11. Equation 5.11 is different from Equation 5.10 in this way: in Equation 5.10, the shared feature set Z is the bridge connecting the target data and auxiliary data; while in Equation 5.11, a subset of target data is the bridge connecting features. We apply the self-taught clustering algorithm in this task to minimize Equation 5.11 through optimizing this objective function into the form of Kullback-Leibler divergence [18] (KL divergence), and then minimize the reformulated objective function.

Finally, if we iteratively choose the best cluster $\tilde{y}$ for each y to minimize $D(p(X|y)||\tilde{p}(X|\tilde{y}))$, the objective function 5.11 will be minimized monotonically. Formally,

$$C_Y(y) = \arg\min_{\tilde{y}\in\tilde{Y}} D(p(X|y)||\tilde{p}(X|\tilde{y})) \tag{5.12}$$

Using a similar argument on Z and X, we have

$$C_Z(z) = \arg\min_{\tilde{z}\in\tilde{Z}} D(q(X|z)||\tilde{q}(X|\tilde{z})) \tag{5.13}$$

$$\begin{aligned} C_X(x) = \arg\min_{\tilde{x}\in\tilde{X}} & p(x)D(p(Y|x)||\tilde{p}(Y|\tilde{x})) \\ & +\lambda q(x)D(q(Z|x)||\tilde{q}(Z|\tilde{x})) \end{aligned} \tag{5.14}$$

In each iteration, the optimization algorithm minimizes the objective function by choosing the best $\tilde{y}$, $\tilde{z}$ and $\tilde{x}$ for each y, z and x based on Equation 5.12, 5.13 and 5.14, respectively. As discussed in [22], this can reduce the value of the global objective function in Equation 5.11.

- Learning Relation Clustering Function

Subsequently, with map functions C_X, C_Y and C_Z on X, Y and Z, we are interested in finding map function $C_{X_{all}}$ for the whole target data X_{all}. Let F be a discrete

random variable, taking values from the whole feature space $Y \bigcup Z$. Similar to learning functions for feature clustering, the final objective function defined as minimizing loss in mutual information between entity pairs and features can be formulated as

$$\begin{aligned} & I(X_{all}, F) - I(\tilde{X_{all}}, \tilde{F}) \\ & \quad = D(p(X_{all}, F)||\tilde{p}(X_{all}, F)) \end{aligned} \tag{5.15}$$

From the same induction as for the objective loss function for feature clustering, to minimize Equation 5.15 is to reduce the value of $D(p(X_{all}, F)||\tilde{p}(X_{all}, F))$.

We have

$$\begin{aligned} & D(p(X_{all}, F)||\tilde{p}(X_{all}, F)) \\ & \quad = \sum_{\tilde{x} \in \tilde{X_{all}}} \sum_{\tilde{f} \in \tilde{F}} \sum_{x \in \tilde{x}} \sum_{f \in \tilde{f}} p(x, f) log \frac{p(x, f)}{\tilde{p}(x, f)} \end{aligned} \tag{5.16}$$

Since $\tilde{p}(x, f) = p(x) \frac{p(\tilde{x}, \tilde{f})}{p(\tilde{x})} \frac{p(f)}{p(\tilde{f})}$, we have

$$\begin{aligned} & D(p(X_{all}, F)||\tilde{p}(X_{all}, F)) \\ & = \sum_{\tilde{x} \in \tilde{X_{all}}} \sum_{\tilde{f} \in \tilde{F}} \sum_{x \in \tilde{x}} \sum_{f \in \tilde{f}} p(x) p(f, x) log \frac{p(x) p(f/x)}{p(x) \tilde{p}(f/\tilde{x})} \\ & = \sum_{\tilde{x} \in \tilde{X_{all}}} \sum_{x \in \tilde{x}} p(x) \sum_{\tilde{f} \in \tilde{F}} \sum_{f \in \tilde{f}} p(f/x) log \frac{p(f/x)}{\tilde{p}(f/\tilde{x})} \end{aligned}$$

where $\tilde{X_{all}}$ is the objective cluster set, Y and Z are independent, we have

$$\begin{aligned} & D(p(X_{all}, F)||\tilde{p}(X_{all}, F)) \\ & \quad = \sum_{\tilde{x} \in \tilde{X_{all}}} \sum_{x \in \tilde{x}} p(x) \{ \sum_{\tilde{y} \in \tilde{Y}} \sum_{y \in \tilde{y}} p(y/x) log \frac{p(y/x)}{\tilde{p}(y/\tilde{x})} \\ & \qquad\qquad + \lambda \sum_{\tilde{z} \in \tilde{Z}} \sum_{z \in \tilde{z}} p(z/x) log \frac{p(z/x)}{\tilde{p}(z/\tilde{x})} \} \\ & \quad = \sum_{\tilde{x} \in \tilde{X_{all}}} \sum_{x \in \tilde{x}} p(x) \{ D(p(Y|x)||p(Y|\tilde{x})) \\ & \qquad\qquad + \lambda D(p(Z|x)||p(Z|\tilde{x})) \} \end{aligned}$$

Since Web features and linguistic features have been clustered, $\tilde{X}$ is used as the seed cluster set, if we choose the best cluster $\tilde{x}$ from $\tilde{X}$ for each x in $X_{all} - X$ to

minimize $D(p(X,F)||\tilde{p}(X,F))$, the objective function will be minimized. Formally

$$C_X(x_{all}) = \tilde{x}, x_{all} \in X \& x_{all} \in \tilde{x} \tag{5.17}$$

$$\begin{aligned} & C_{X_{all}}(x_{all}) \\ & = \arg\min_{\tilde{x}\in\tilde{X}}\{p(x_{all})D(p(Y|x_{all})||\tilde{p}(Y|(\tilde{x})) \\ & +\lambda q(x_{all})D(q(Z|x_{all})||\tilde{q}(Z|(\tilde{x}))\}, x_{all} \notin X \end{aligned} \tag{5.18}$$

Based on Equation 5.17 and 5.18, an alternative way to minimize the objective function in Equation 5.15 is derived. If entity pair x is in the common data X, we simply choose the cluster $\tilde{x}$ that it maps to using map function C_X.

5.4 Experiments

In this section, we evaluate our multi-view co-clustering approach on the relation extraction task, and show the effectiveness of the proposed approach.

5.4.1 Experimental Setup

We conduct our experiments on relation extraction task using the dataset that was created for evaluating relation extraction from Wikipedia in [61]. The dataset consists of 3833 positive relation instances (entity pairs), for 13 relation types which are the Spouse, President, Vice_Chairman, COO, Director, Chairman, Founder, CEO, Birth_date, Birth_place, Products, Foundation and Location relations. Each relation instance (entity pair) in the dataset has one accompanying sentence from a Wikipedia article.

We build two baseline systems on the dataset. One baseline system is built using [69]'s URI method which showed that their algorithm improved over previous work using Web features for unsupervised relation extraction: features that test two entities together and features that test only one slot each. We use this system to represent the performance of Web-based relation extraction methods. The other system is built using [15]'s method, which is demonstrated in their paper, that outperforms other clustering

Table 5.1: Performance comparison using different methods

Relation	Linguistic feature clustering			Web feature clustering			Multi-view clustering		
	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.
Spouse	19.02	45.13	26.76	52.31	39.73	45.16	64.23	51.58	57.21
President	14.07	40.00	20.82	19.71	25.00	22.04	22.63	39.51	28.78
Vice_Chairman	67.14	14.81	24.27	20.61	16.67	18.43	45.82	26.64	33.69
COO	100.0	11.17	20.10	14.55	10.88	12.45	25.78	21.42	23.40
Director	87.50	42.31	57.04	40.25	37.69	38.93	55.32	47.57	51.15
Chairman	24.62	21.59	23.01	41.79	43.54	42.65	57.36	46.45	51.33
Founder	72.70	59.43	65.40	28.99	52.61	37.38	67.02	71.49	69.18
CEO	48.89	17.49	25.76	35.96	42.62	39.01	51.85	41.90	46.35
Birth_date	56.67	72.35	63.56	73.80	82.06	77.71	78.62	88.74	83.37
Birth_place	24.93	13.19	17.25	63.19	48.70	55.01	66.31	51.57	58.02
Products	100.0	11.16	20.08	58.67	31.32	40.84	63.51	36.14	46.07
Foundation	72.26	53.42	61.43	61.11	47.83	53.66	84.32	63.86	72.68
Location	72.16	16.97	27.48	63.91	51.82	57.23	74.19	49.86	59.64
overall	41.18	31.47	35.68	47.31	45.72	46.50	67.74	54.03	60.11

Table 5.2: Overall performance

Method	Pre.	Rec.	F-v.
Linguistic clustering	41.18	31.47	35.09
Web clustering	47.31	45.72	46.50
Proposed clustering	67.74	54.03	60.11

methods by use of various lexical and syntactic features from the contexts. We use it to represent the performance of linguistic-based relation extraction methods.

To evaluate the performance of our approach, we collect Web features through querying with entity pairs by a search engine (Google). Different from simply taking the entire string between two concept words which capture an excess of extraneous and incoherent information, our idea of getting Web features is to look for verbs, nouns, prepositions, and coordinating conjunctions that can help make explicit the hidden relations between the target nouns. To collect linguistic features, for each entity pair, the accompanying sentence is parsed using a linguistic parser. We generate dependency patterns as sub-paths from the shortest dependency path [61] containing two entities by making use of a frequent tree-mining algorithm [87].

In these experiments, we use precision, recall, and F-value to measure the performance of different methods. The following quantities are considered to compute precision, recall, and F-value:

- p = the number of detected entity pairs.
- p' = the number of detected entity pairs which are actual relation instances.
- n = the number of actual relation instances.

$$\text{Precision } (Pre.) = \text{p'/p} \quad \text{Recall } (Rec.) = \text{p'/n}$$
$$F\text{-value } (F - v) = 2 * Pre. * Rec./(Pre. + Rec.)$$

5.4.2 Empirical Analysis

Table 5.1 presents the comparison between our approach and two baseline systems. Using our multi-view co-clustering approach, it is effective to integrate Web features and linguistic features by information transformation among Web features, linguistic

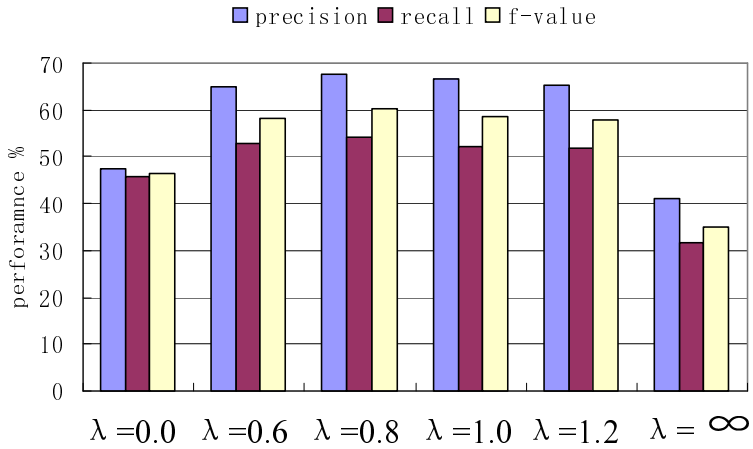

Figure 5.3: Performance against trade-off values

features and entity pairs, with precision 67.74%, recall 54.03% and F-value 60.11%. From this table, we can see that the performance of the proposed approach is better than both the Web-based method (with precision 47.31%, recall 45.72% and F-value 46.50%) and the linguistic-based method (with precision 41.18%, recall 31.47% and F-value 35.68%) for relation extraction task. Using different feature sets, the performance is different. It shows that each kind of feature type contributes differently to our task. Another observation is that Web features and linguistic features provide complementary information to relation extraction task, so that by learning the connectivity between them, the performance of relation extraction is boosted. It's worth noting that our multi-view co-clustering approach shows much higher precision than both Web-based and linguistic-based methods, with similar features clustered into small groups and by minimizing loss in mutual information before and after clustering of Web features, linguistic features and entity pairs.

We use the feature clustering function described in section 3.3.1 to cluster Web features and linguistic features, and use relation clustering function described in section 3.3.2 to cluster entity pairs. As described in Equation 5.11 and 5.18, a trade-off parameter λ between Web and linguistic features is used to determine the contribution of different features. As shown in Figure 5.3, we test the dataset against several values of λ: $\lambda = 0.6$, $\lambda = 0.8$, $\lambda = 1.0$ and $\lambda = 1.2$. $\lambda = 0.0$ means using only

Table 5.3: Most frequent Web features in the clusters

Spouse	X marry Y	X be married to Y	X wife Y	X husband Y
President	X president of Y	X be president of Y	X president for Y	Y president X
Vice_Chairman	Y vice chairman X	X be vice chairman of Y	X as Vice Chairman of Y	X vice chairman of Y
COO	Y coo X	Y be coo of X	X be chief operating officer of Y	X as coo of Y
Director	X be director of Y	X Y director	X director Y	X director of Y
Chairman	X be chairman of Y	X ceo and chairman of Y	Y chairman of committee X	Y board chairman X
Founder	Y be found by X	X founder Y	X be founder of Y	Y founder X
CEO	X be ceo of Y	Y ceo X	X be chief executive officer of Y	Y ceo X
Birth_date	X be bear on Y	X bear in Y	X bear in Y	bear in Y X
Birth_place	X be bear in Y	X be bear in district of Y	X birth place Y	X birthplace Y
Products	X supplier of product to Y	X deliver Y	X launch Y	X provide Y service
Foundation	X be find in Y	X based in Y	X establish in Y	X foundation Y
Location	X located in Y	X be located in Y	X be headquartered in Y	X site in Y

Table 5.4: Most frequent linguistic features in the clusters

Spouse	(marry(subj:)(obj:(Y)))	(marry(subj:(X))(obj:(Y)))	(married(v-ch:(to(pcomp:))))
President	(president(mod:(of)))	(president(mod:))	(president(mod:(of(pcomp:))))
Vice_Chairman	(vice-chairman(mod:(of)))	(vice-chairman(mod:(of)))	(be(vice-chairman(mod:)))
COO	(coo(ha:(of(pcomp:))))	(coo(ha:(of(pcomp:(Y))))	(coo(ha:(of)))
Director	(be(comp:(director)))	(director(mod:(of(pcomp:))))	(be(subj:)(comp:(director(mod:))))
Chairman	(be(comp:(chairman)))	(chairman(mod:(of(pcomp:(Y)))))	(become(subj:)(comp:(chairman)))
Founder	(found(agt:(by)))	(found(agt:(by(pcomp:(X)))))	(co-founder(mod:(of(pcomp:))))
CEO	(become(comp:(ceo)))	(become(comp:(ceo(mod:(of)))))	(X(attr:(ceo)))
Birth_date	(bear(v-ch:)(ha:(Y)))	(bear(v-ch:(be(subj:)))(tmp:(in)))	(bear(v-ch:(be(subj:(X))))
Birth_place	(bear(loc:))	(bear(v-ch:(be(subj:)))(loc:(in)))	(bear(v-ch:(be))(loc:))
Products	(provide(subj:(X)))	(provide(subj:(X))(obj:(Y)))	(provider(mod:(include(obj:(Y)))))
Foundation	(found(loc:(in)))	(form(v-ch:(be(subj:))))	(establish(phr:(in(pcomp:(Y)))))
Location	(X(mod:(locate(loc:))))	(locate(loc:(in)))	(base(loc:(in(pcomp:(Y)))))

Web features, while $\lambda = \propto$ means using only linguistic features. It can be seen that the performance is the best when λ is 0.8. This means that Web features contribute more than linguistic features. The results support our assumptions about Web information and linguistic analysis technologies: 1) Dependency analysis can abstract away from different surface realizations of text. In addition, embedded structures of the dependency representation are important features for relation extraction task. 2) Surface patterns are used to merge concept pairs with relations represented in different dependency structures with redundancy information from the vast size of Web pages. Using surface patterns, more concept pairs are clustered, and the coverage is improved.

For each relation cluster in Table 5.3, we show top four Web features that occur with the largest frequency. From Table 5.3, it is clear that each cluster contains different Web features that express a specific semantic relation. X and Y in feature

expressions are used to label the first entity and second entity of a relation instance respectively. Similarly, in Table 5.4, for each relation cluster, we show the top three linguistic features that occur with the largest frequency. We see that linguistic features in different surface expressions are clustered to represent the same semantic relation. Moreover, each cluster contains different linguistic features that express a specific semantic relation. Each linguistic feature denotes one tree transaction represented in strict S-expression. Strict means that all nodes, even leaf nodes, must be bracketed.

All the experimental results support our idea mainly in two main ways: 1) the combination of Web features and linguistic features is effective in relation extraction task; 2) multi-view co-clustering learning which makes use of knowledge gained from feature learning task is feasible to improve the performance of relation clustering task even in an unsupervised way.

5.5 Conclusions

To discover a range of semantic relationships from large-scale corpus, we present an unsupervised relation extraction approach to use deep linguistic information to alleviate surface and noisy surface features generated from large corpus, and use Web frequency information to ease the sparseness of linguistic information. We propose a multi-view co-clustering approach for semantic relation extraction task. One is learning clustering functions for Web features and linguistic features simultaneously. The other is learning a clustering function for entity pairs based on feature clustering functions. The proposed approach is an instance of unsupervised multi-view clustering. To the best of our knowledge, our approach is novel for various machine learning applications, especially for semantic relation extraction task. We report our experimental results comparing it to previous work and evaluating it over using different features. The results show that the performance of our proposed approach is the best when compared with several existed clustering methods.

Chapter 6

Multi-view Bootstrapping Approach by Exploring Web Features and Linguistic Features

In this chapter, we further study the open question of chapter 4: how to use the results of unsupervised clustering to harvest a large number of instances of these relations for Wikipedia concepts. We propose to use the clustering results for seeding a semi-supervised relation extraction system for bootstrapping a high-recall relation extraction process. another multi-view method by integrating frequency information from Web with linguistic analysis on Wikipedia texts for our open relation extraction from Wikipedia. With a novel view on integrating syntactic analysis on Wikipedia text with redundancy information from the Web, we propose a multi-view learning approach for bootstrapping relationships between entities with the complementary between the Web view and linguistic view. On the one hand, from the linguistic view, linguistic features are generated from linguistic parsing on Wikipedia texts by abstracting away from different surface realizations of semantic relations. On the other hand, Web features are extracted from the Web corpus to provide frequency information for relation extraction.

6.1 Introduction

Recent attention to automatically harvesting semantic resources from Wikipedia has encouraged Data Mining researchers to develop algorithms for it. Many efforts have been focused on extracting semantic relations between entities, such as *birth_date* relation, *CEO* relation, and other relations.

Currently one type of the leading methods in semantic relation extraction are based on collecting relational frequency patterns or terms from a local corpus or use the Web as corpus [66]; [58]; [4]; [24]; [8]. Let us call them frequent pattern mining-based methods. The standard process is to scan or search the corpus to collect co-occurrences of word pairs with strings between them, then from collective strings calculate term co-occurrence or generate textual patterns. In order to clearly distinguish from linguistic features below, let us call them Web features. For example, given an entity pair $< x, y >$ with $Spouse$ relation, string "*x is married to y*" is a textual pattern example. The method is used widely, however, frequent pattern mining is non-trivial since the number of unique patterns is loose but many are non-discriminative and correlated. One of the main challenges and research interest for frequent pattern mining is how to abstract away from different surface realizations of semantic relations to discover discriminative patterns efficiently.

Another type of leading methods are using linguistic analysis for semantic relation extraction from well-written texts(see e.g., [47]; [11]; [89]). Let us call them syntactic analysis-based methods. Currently, syntactic analysis-based methods for semantic relation extraction are almost all supervised, relying on pre-specification of the desired relationship or hand-coding initial training data. The main process is to generate linguistic features based on the analysis of the syntactic, dependency or shallow semantic structure of text, then through training to identify entity pairs which assume a relationship and classify them into pre-defined relationships. For example, given an entity pair $< x, y >$ and the sentence "*x is the wife of y*", syntactic, dependency features will be generated by analysis of the sentence. One of the main disadvantages is that semantic relations maybe expressed in different dependency/syntactic structures. Moreover, for the heterogeneous text found on the Web, it often runs into problems to apply "deep" linguistic technology.

Syntactic analysis-based methods extract relation instances with similar linguistic features to abstract away from different surface realizations of semantic relations, while frequent pattern mining-based methods group different surface patterns for one relation instance from redundancy Web information are expected to address the data sparseness problem. Wikipedia, unlike the whole Web corpus, as an earlier report [37] explained, Wikipedia articles are much cleaner than typical Web pages, we can use "deep" linguistic technologies, such as syntactic or dependency parsing. Considering the complementary of the strengths and the weaknesses of both two views, we propose a multi-view learning approach for relation extraction from Wikipedia with view disagreement detection which can be advantageous when compared to learning with only a single view. To decide whether two relation instances share the same relationship, a common assumption in multi-view learning is that the samples from each view always belong to the same class. In realistic settings, linguistic-view and Web-view are often corrupted by noise. For example, it happens that dependency parsing for some long sentences will be erroneous. Thus we also consider filtering view corruption which is a source of view disagreement.

In this chapter we present a method for performing multi-view learning by filtering view disagreement between linguistic features and Web features. We learn a classifier in a bootstrapping way for each relation type from confident trained instances with view disagreement detected by exploiting the joint view statistics.

The main contributions of this part of work are as follows:

- With a novel view on integrating linguistic analysis on Wikipedia text with redundancy information from the Web, we propose a multi-view learning approach for bootstrapping relationships between entities with the complementary between the Web view and linguistic view. From the Web view, related information between entity pairs are collected from the whole Web. From linguistic view, syntactic and dependency information are generated from appropriate Wikipedia sentences.

- Different from traditional multi-view learning approaches for relation extraction task, we filter view disagreement to deal with view corruption between linguistic features and Web features, only confident instances without view disagreement are used to bootstrap learning relations.

- Our study suggests an example to bridge the gap between Web mining technology and "deep" linguistic technology for information extraction tasks. It shows how "deep" linguistic features can be combined with features from the whole Web corpus to improve the performance of information extraction tasks. And we conclude that learning with linguistic features and Web features is advantageous comparing to only one view of features.

The remainder of the chapter is organized as follows. In section 6.2 we will consider related work of this work. In section 6.3 we present out our approach. In section 6.4 we will report on our experimental results. Finally, in section 6.5 we will conclude the chapter.

6.2 Related Work

In this section, we review several past research works that are related to our work, including, frequency pattern mining-based relation extraction, syntactic analysis-based relation extraction and multi-view bootstrapping methods.

The World Wide Web is a vast resource for information. Snowball[1] introduced strategies for generating patterns and extracting tuples from plain-text documents that required only a handful of training examples from users. At each iteration of the extraction process, Snowball evaluated the quality of these patterns and tuples without human intervention, and kept only the most reliable ones for the next iteration. [58] extracted underlying relations among entities from social networks (e.g., person-person, person-location net- work). They obtained a local context in which two entities co-occur on the Web, and accumulated the context of the entity pair in different Web pages. They defined the context model as a vector of terms surrounding the entity pair. [8] proposed a relational similarity measure, using a Web search engine, to compute the similarity between semantic relations implied by two pairs of words. They represented various semantic relations that exist between a pair of words using automatically extracted lexical patterns. The extracted lexical patterns were then clustered to identify the different patterns that expressed a particular semantic relation. In this work, motivated by the work of [58] and [8], we extract relational terms and textual pattern from Web contexts as Web view.

Currently syntactic analysis-based relation extraction approaches for semantic relation extraction are almost supervised. Many methods, such as feature-based [47]; [91], tree kernel- based ([88]; [20]) and composite kernel-based ([90]; [89], have been proposed in literature. Zhang et al. (2006)[89] presented a composition kernel to extract relations between entities with both entity kernel and a convolution parse tree kernel. As show in their paper, composition of entity features and structured features outperforms using only one kinds of features. Their work also suggests that structured syntactic information has good predication power for relation extraction and the structured syntactic information can be well captured by the tree kernel. This indicates that the flat and the structured features are complementary and the composite of features is effective for relation extraction. Motivated by the work of (Zhang et al., 2006), we here generate entity features and tree sub-structure features as linguistic view.

Multi-view learning approaches form a class of semi-supervised learning techniques that use multiple views to effectively learn from partially labeled data. [6] introduced co-training which bootstraps a set of classifiers from high confidence labels. [19] proposed a co-boost approach that optimizes an objective that explicitly maximizes the agreement between each classifier, while [72] defined a co-regularization method that learns a multi-view classifier from partially labeled data using a view consensus-based regularization term. [17] have reported a filtering approach to handle view disagreement, and developed a model suitable for the case where the view corruption is due to a background class.

In this study, we propose a multi-view bootstrapping approach for relation extraction from linguistic and Web views. On the one hand, from the Web view, Web features are generated from the Web redundancy information to provide frequency information. On the other hand, from the linguistic view, syntactic features are generated from Wikipedia sentences by linguistic analysis to abstract information away from surface realizations of texts. Our approach bootstrap learns a classifier for each relation type from confident trained instances by applying Christoudias et al. [17]'s view disagreement detection strategy.

6.3 Multi-view Bootstrapping

We propose a multi-view bootstrapping approach for relation extraction from Wikipedia based on two views of features - Web features and linguistic features - with view agreement detection strategy.

6.3.1 Outline of the Proposed Method

The proposed method consists of three steps. In this section, we give a brief overview of each of those steps. The subsequent sections will explain the steps in detail.

Let us assume that we are given a set of entity pairs (X, Y), the task is to classify all entity pairs into several groups, each of which represent a pre-specified semantic relationship. We first query a Web search engine to find the contexts in which the two entity words co-occur, and extract Web features that express semantic relations between the entity pair. Then we select sentences containing both entity words from Wikipedia articles, generate linguistic features such as dependency sub-structures by parsing the selected sentences using a linguistic parser. Next, since there can be more than one features that express the same semantic relation, we cluster the features to identify the ones that express a particular semantic relation. Finally, we present a multi-view bootstrapping method that learns from confident instances with view disagreement detection.

The approach consists of three steps:

- Step1: Feature Acquisition. For each entity pair, generates linguistic features from corresponding Wikipedia texts using linguistic analysis and extracts Web features from context information by searching the Web.
- Step2: Feature Clustering. Clusters Web feature and linguistic features respectively to identify the ones that express a particular semantic relation. We cluster features to avoid computing the similarities of features during the bootstrapping.
- Step3: Multi-View Bootstrapping. For each relation type, learns a classifier which initially trained from a seed set. During bootstrapping, confidently classified samples in each view are used to label instances in the other views.

6.3.2 Feature Acquisition

For each entity pair, we generate two kinds of features: linguistic features from Wikipedia texts through linguistic analysis and Web features by searching context information from the Web.

6.3.2.1 Web Feature Generation

Querying an entity pair using a search engine (e.g. Yahoo!), we characterize the semantic relation between the pair by leveraging the vast size of the Web. Our hypothesis is that there exist some key terms and patterns that provide clues to the relations between entity pairs. From the snippets retrieved by the search engine, we extract relational information of two kinds: ranked relational terms as keywords and surface patterns.

- Relational Terms Collection

To collect relational terms as indicators for each entity pair, we look for verbs and nouns from qualified sentences in the snippets instead of simply finding verbs. Using only verbs as relational terms might engender the loss of various important relations, e.g. noun relations "CEO", "founder" between a person and a company. Therefore, for each concept pair, a list of relational terms is collected. Then all the collected terms of all concept pairs are combined and ranked using an entropy-based algorithm which is described in [14]. With their algorithm, the importance of terms can be assessed using the entropy criterion, which is based on the assumption that a term is irrelevant if its presence obscures the separability of the dataset. After the ranking, we obtain a global ranked list of relational terms T_{all} for the whole dataset (all the entity pairs). For each entity pair, a local list of relational terms T_{ep} is sorted according to the terms' order in T_{all}. Then from the relational term list T_{ep}, a keyword t_{ep} is selected for each entity pair *ep* as the first term appearing in the term list T_{ep}. t_{ep} will be used to generate surface patterns below.

- Surface Pattern Generation

Because simply taking the entire string between two entity words captures an excess of extraneous and incoherent information, we use T_{ep} of each entity pair as a key

Table 6.1: Surface pattern samples for an entity pair

Pattern	Pattern
ep ceo *es*	*es* found *ep*
ceo *es* found *ep*	*es* succeed as ceo of *ep*
es be ceo of *ep*	*ep* ceo of *es*
ep assign *es* as ceo	*ep* found by ceo *es*
ceo of *ep es*	*ep* found in by *es*

for surface pattern generation. We classified words into Content Words (CWs) and Functional Words (FWs). From each snippet sentence, two entity words and the keyword t_{ep} is considered to be a Content Word (CW). Our idea of obtaining FWs is to look for verbs, nouns, prepositions, and coordinating conjunctions that can help make explicit the hidden relations between the target nouns.

Surface patterns have the following general form.

$$[\textbf{CW1}]\ Infix_1\ [\textbf{CW2}]\ Infix_2\ [\textbf{CW3}] \tag{6.1}$$

Therein, $Infix_1$ and $Infix_2$ respectively contain only and any number of FWs. A pattern example is "*ep* assign *ep* as *ceo (keyword)*". All generated patterns are sorted by their frequency, and all occurrences of the principle entity and the second entity are replaced with "*ep*" and "*es*", respectively for pattern matching of different entity pairs.

Table 6.1 presents examples of surface patterns for a sample entity pair. Pattern windows are bounded by CWs to obtain patterns more precisely because 1) if we use only the string between two entity words, it may not contain some important relational information, such as "ceo *ep* resign *es*" in Table 6.1; 2) if we generate patterns by setting a windows surrounding two entity words, the number of unique patterns is often exponential.

6.3.2.2 Linguistic Feature Extraction

We select sentences from Wikipedia articles containing both entities. We define the composite feature vector with flat and the structured features generated from these sentences by using a syntactic parser.

- Flat Features

Using a syntactic parser (Connexor[1]), rich linguistic tags can be extracted as features for each entity in an entity pair. For each pair of entities, we extract the following syntactic features as flat features:

- Morphology Features: tells the details of word forms used in text. Connexor Parser defines 70 morphology tags such as *N*(noun), *NUM* (numeral) .
- Syntax Features: describes both surface syntactic and syntactic function information of words. For example, *%NH* (nominal head) and *%>N* (determiner or premodifier of a nominal) are surface syntactic tags, *@SUB* (Subject) and *@F-SUBJ* (Formal subject) are syntactic function tags.
- Structure Features

To obtain structured features for an entity pair, we generate dependency patterns. After preprocessing, selected sentences that contain at least one mention of both entity words are parsed into dependency structures. We define dependency patterns as sub-paths of the shortest dependency path between an entity pair for two reasons. One is that the shortest path dependency kernels outperform dependency tree kernels by offering a highly condensed representation of the information needed to assess their relation [11]. The other reason is that embedded structures of the linguistic representation are important for obtaining good coverage of the pattern acquisition, as explained in [20]; [89]. The process of inducing dependency patterns has two steps, as shown in Fig. 6.1.

1. Shortest dependency path inducement. From the original dependency tree structure by parsing the selected sentence for each entity pair, we first induce the shortest dependency path from the Wikipedia sentence with the pair of entity words, as shown in the left side of Fig. 6.1.

2. Dependency pattern generation. We use a frequent tree-mining algorithm [87] to generate sub-paths as dependency patterns from the shortest dependency path, as shown in the right side of Fig. 6.1.

[1] www.connexor.com

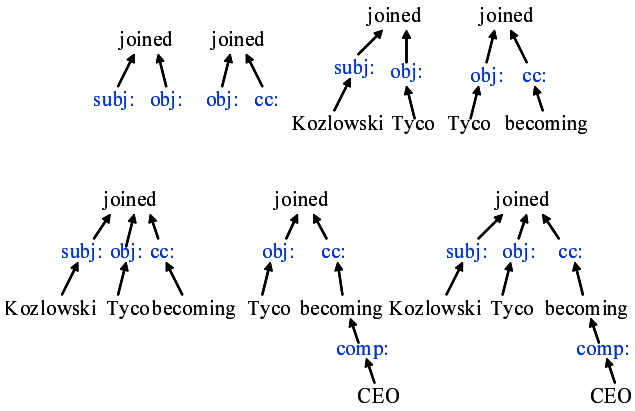

Figure 6.1: Example showing how to generate dependency patterns for an entity pair

6.3.3 Feature Clustering

A semantic relation can be expressed using more than one pattern. When we compute the relational similarity between two entity pairs, it is important to know whether there is any correspondence between the sets of patterns extracted for each entity pair. If there are many related patterns between two entity pairs, we can expect a high relational similarity. To find semantically related lexical patterns for each view, we apply Sequential pattern clustering algorithm in [8] by using distributional hypothesis [42]. Distributional hypothesis claims that words that occur in the same context have similar meanings.

Given a set P of patterns and a clustering similarity threshold , their algorithm returns clusters (of patterns) that express similar semantic relations. First, their algorithm sorts the patterns into descending order of their total occurrences in all word pairs. Next, it repeatedly takes a pattern p_i from the ordered set P, finds the cluster that is most similar to p_i. To compute the similarity between a pattern and a cluster, first they represent a cluster by the vector sum of all entity pair frequency vectors corresponding to the patterns that belong to that cluster. Next, they compute the cosine of the angle between the vector that represents the cluster (c_j), and the word-pair frequency vector of the pattern (p_i). The sequential nature of their algorithm avoids pairwise comparisons among all patterns. Moreover, sorting the patterns by their total word-pair frequency prior to clustering ensures that the final set of clusters contains

the most common relations in the data-set.

6.3.4 Multi-View Bootstrapping with View Disagreement Detection

In this section we present a multi-view bootstrapping algorithm that uses the idea of view disagreement detection. We apply (Christoudias, et al., 2008)[17]'s conditional view entropy measure to detect and filter entity pairs with view disagreement in a pre-filtering step.

Multi-view learning can be advantageous when compared to learning with only a single view especially when the weaknesses of one view complement the strengths of the other. A common assumption in multi-view learning is that the samples from each view always belong to the same class. In realistic settings, datasets are often corrupted by noise. Thus we need to consider view disagreement caused by view corruption. We apply the method in (Christoudias, et al., 2008)[17] for Multi-view Bootstrapping by learning a classifier in one view from the labels provided by a classifier from another view with a view disagreement strategy. Their Method consists of two steps:

- Step1: View disagreement detection. Detect and filter entity pairs with view disagreement using an information theoretic measure based on conditional view entropy.
- Step2: Multi-view Bootstrapping. Mutually train a set of classifiers, on an unlabeled dataset by iteratively evaluating each classifier and re-training from confidently classified entity pairs.

Firstly, to detect view disagreement, they use conditional entropy $H(x|y)$ which is a measure of the uncertainty in x given that we have observed y. In the multi-view setting, the conditional entropy between views, $H(x_i|x_j)$, can be used as a measure of agreement that indicates whether the views of a sample belong to the same class or event. Under the assumptions the conditional view entropy is expected to be larger when conditioning on entity pairs with disagreement compared to those without disagreement. When computing the conditional entropy between views, we use the pattern clusters to replace features when measuring the conditional entropy between views so we can avoid computing the distance between two similar patterns.

Table 6.2: Overview of the dataset

relation	#Instance	Instance samples for each relation type
job_title	216	(Charles Darwin, naturalist), (Jack Kerouac, novelist)
birth_year	157	(Hillary Clinton, 1947), (George H. W. Bush, 1924)
education	106	(J. Bowdoin, Harvard), (F. Schaffner, Columbia University)
death_year	104	(Abraham Lincoln,1865), (James Bowdoin, 1790)

Secondly, with the conditional entropy measure, we mutually train a set of classifiers for each relation type, on an unlabeled dataset iteratively evaluating each classifier and re-training from confidently classified samples. In the presence of view disagreement, we detect classified samples which are not in view disagreement. Only those detected classified samples are used to train classifiers iteratively. During bootstrapping, confidently classified samples in each view are used to label corresponding samples in the other views.

6.4 Experiments

In this section, we evaluate our multi-view bootstrapping approach on the relation extraction from Wikipedia, and show the effectiveness of the proposed approach.

6.4.1 Experimental Setup

We conduct our experiments on relation extraction task using the dataset that was created for evaluating relation extraction from Wikipedia in [21]. This data contains Wikipedia pages for which links between pages have been annotated with a relation type, e.g. birth_year, education, nationality, etc. We evaluate on a subset which consists of four relation types $job\_title, birth\_year, education, death\_year$. For each relation type, in Table 6.2, we show some of the instances and the total number of entity pairs. Each entity pair in the dataset has one accompanying sentence from a Wikipedia article.

We build four baseline systems on the dataset. The first baseline system is built by bootstrapping from only the linguistic view which shows the performance of learning with only linguistic features. The second system is built by bootstrapping from only

the Web view which shows the performance of learning with Web features. The third baseline system is built by bootstrapping with all the Web and linguistic features in a single view which shows the performance of learning with the combination of Web and linguistic features. We also evaluate on bootstrap learning from the linguistic view and Web view without view disagreement detection in a traditional way.

To evaluate the performance of our approach, we run the feature generation algorithm described in section 6.3.2 for each entity pair in our dataset to extract Web features and linguistic features. We collect Web features through querying with each pair of entity words by a search engine (We use Yahoo, the top 1000 snippets are downloaded as collective context). We collect relational terms and textual patterns as Web features by look for verbs, nouns, prepositions, and coordinating conjunctions that can help make explicit the hidden relations between the target nouns. To collect linguistic features, for each entity pair, the accompanying sentence is parsed by a linguistic parser. We collect entity features for each entity word and generate dependency patterns as sub-paths from the shortest dependency path containing two entities by making use of a frequent tree-mining algorithm [87].

In these experiments, we use precision, recall, and F-value to measure the performance of different methods. The following quantities are considered to compute precision, recall, and F-value:

- p = the number of detected entity pairs.
- p' = the number of detected entity pairs which are actual relation instances.
- n = the number of actual relation instances.

$$\text{Precision } (Pre.) = \text{p'/p} \quad \text{Recall } (Rec.) = \text{p'/n}$$
$$F\text{-value } (F - v.) = 2 * Pre. * Rec./(Pre. + Rec.)$$

6.4.2 Feature Clusters

We use the clustering algorithm described in Section 6.3.3 to cluster the extracted Web features and linguistic features respectively.

For each relation cluster in Table 6.3, we show top four Web features that occur with the largest frequency. From Table 6.3, it is clear that each cluster contains different

Table 6.3: Examples of frequent Web features from Web feature clustering

ep was a es	ep was elected es	ep was the es	ep was the leading es
ep was born in es	ep born in es	ep born D es	ep was born on es
es graduate ep	ep graduated from es	ep is a graduate of es	ep graduated from the es
ep died es	ep died in D es	ep who died in es	ep who died in D es

Table 6.4: Examples of frequent features from linguistic feature clustering

(be(ep))	(be(es))	(mainroot(be(es)))	(be(ep)(es))
(bear(die))	(bear(be)(die))	(mainroot(bear(die)))	(bear(be(ep))(die))
(graduate(ep))	(mainroot(graduate(ep)))	(mainroot(graduate))	(graduate(ep)(from))
(attend(ep))	(attend(ep)(es))	(mainroot(attend(ep)(es)))	(mainroot(attend(ep)))
(bear(es))	(bear(be)(in))	(bear(be(ep)))	(bear(in))

Web features that express a specific semantic relation. ep and es in feature expressions are used to label the first entity and second entity of a relation instance respectively. Similarly, in Table 6.4, for each relation cluster, we show the top four linguistic features that occur with the largest frequency. We see that linguistic features in different surface expressions are clustered to represent the same semantic relation. Moreover, each cluster contains different linguistic features that express a specific semantic relation. Each linguistic feature denotes one tree transaction represented in strict S-expression. Strict means that all nodes, even leaf nodes, must be bracketed.

6.4.3 Empirical Analysis

Table 6.5 presents the overall evaluation of the comparison of our approach and three baseline systems. The first three columns of results show bootstrapping with only one view of features respectively: linguistic view, Web view, the combination view. It

Table 6.5: Overall evaluation over different methods

	Single-View Bootstrapping			Multi-View Bootstrapping	
	Linguistic Feature	Web Feature	Combination view	No disa. detection	Disa. detection
Pre.	46.30	51.80	58.04	54.14	68.19
Rec.	40.82	47.00	53.76	51.03	63.95
F-v.	43.39	49.28	55.82	52.54	66.00

Table 6.6: Evaluation on each relation type over single view

Relation	Linguistic-View bootstrapping			Web-View bootstrapping			Combination-view bootstrapping		
	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.
job_title	69.82	54.63	61.30	66.20	21.76	32.75	72.35	51.62	60.25
birth_year	21.43	15.29	17.84	40.00	53.50	45.78	46.52	40.97	43.57
education	56.52	12.26	20.16	52.63	47.17	49.75	59.17	40.68	48.42
death_year	39.52	79.81	52.87	60.78	89.42	72.37	44.58	90.87	59.82
overall	46.30	40.82	43.39	51.80	47.00	49.28	58.04	53.76	55.82

Table 6.7: Evaluation on each relation type over multi-view methods

Relation	Multi-view without view disagreement			Multi-view with view disagreement		
	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.
job_title	69.75	52.31	59.79	91.18	57.41	70.45
birth_year	43.38	37.58	40.27	57.71	64.33	60.84
education	42.39	36.79	39.40	69.57	60.38	64.65
death_year	48.19	89.42	62.63	53.33	92.31	67.61
overall	54.14	51.03	52.54	68.19	63.95	66.00

shows that the performance of using Web features is better than using linguistic features, and with the combination of Web and linguistic features, the performance is even better. It's worth noting that by applying traditional bootstrapping method with Web features and linguistic features without view disagreement detection, the performance is a little worse than bootstrapping with the combination view, while by multi-view learning with view disagreement detection, the performance is boosted greatly. A closer look into the learning process reveals that learning with instances with view corruption from one view is often erroneous to classify instances from the other view. Multi-view bootstrapping learning with view disagreement detection also shows that Web features and linguistic features provide different information for the relation extraction task. It means that by dealing with view corruption, relations can be learned with better reliability from confident samples.

We also compared the above three baseline systems with our proposed method

for the four relation types, shown in Table 6.6 and Table 6.7. Using only linguistic features, the performance is much worse than Web views for some relationships, such as "birth_year". A closer look into the features extracted for some entity pairs reveals that some instances which belong to different relation types are often described in the same Wikipedia sentence. This kind of sentences are often hard to be parsed in an appropriate way to generate the correct linguistic features. For Example, "*Aldous Leonard Huxley (July 26, [[1894]] C November 22, [[1963]]) was a British [[writer]]*" is the Wikipedia sentence containing instances of relations "job_title", "birth_year", "death_year".

All the experimental results support our idea mainly in three main ways: 1) the combination of Web features and linguistic features is effective in relation extraction task; 2) It has been shown that with traditional multi-view bootstrapping, the overall performance is a little lower than bootstrap learning with only a single view with Web and linguistic features due to view corruption. 3) the detection and filtering of view disagreement considerably increases the performance of traditional multi-view learning approaches. It also has been shown that with view disagreement detection, it is also advantageous to learning with only a single view when the weaknesses of one view complement the strengths of the other after the filtering of view corruption.

6.5 Conclusions

We propose a multi-view learning approach for bootstrapping relationships between entities from Wikipedia with the complementary between the Web view and linguistic view. From Web view, related information for entity pairs are collected from the whole Web. From linguistic Web, analysis information from sentences are generated from Wikipedia sentences. We filter view disagreement to deal with view corruption between linguistic features and Web features, with only confident trained instances used for classifiers. Experimental evaluation on a relational dataset demonstrates that linguistic analysis and Web collective information reveal different aspects of the nature of entity-related semantic relationships. Our multi-view learning method considerably boosts the performance comparing to learning with only one view, with the weaknesses of one view complement the strengths of the other. This study suggests an example to

bridge the gap between Web mining technology and “deep” linguistic technology for information extraction tasks.

Chapter 7

Conclusion and Future Work

In this section, we conclude the thesis and describe the future research directions.

7.1 Summary of the Thesis

In this thesis, we systematically studied two types of relation extraction: relation extraction for linguistic parsing and for semantic repository construction. In Chapter 3, facing the challenge of extracting a universal set of semantic or thematic relations covering various types of semantic relationships between entities, based on the Concept Description Language for Natural Language (CDL.nl) which defines a set of semantic relations to describe the concept structure of text, we studied the first type of relation extraction by developing a shallow semantic parser to add a new layer of semantic annotation of natural language sentences as an extension of SRL. We proposed a hybrid relation extraction approach: a rule-based method is presented to detect all entity pairs between each of pair for which there exists a relationship; then, a kernel-based method is proposed to assign a CDL.nl relation to each detected entity pair. Preliminary evaluation on a manual dataset shows that CDL.nl relations can be extracted with good performance.

In Chapter 4, we studied another type of relation extraction: retrieving facts from text to construct semantic resource which are believed to be interested by human users. We presented an open relation extraction method for discovering and enhancing relations in which a specified concept in Wikipedia participates. Using respective characteristics of Wikipedia articles and Web corpus, we develop a clustering approach

based on combinations of patterns: dependency patterns from dependency analysis of texts in Wikipedia, and surface patterns generated from highly redundant information related to the Web. Evaluations of the proposed approach on two different domains demonstrate the superiority of the pattern combination over existing approaches. Fundamentally, our method demonstrates how deep linguistic patterns contribute complementarily with Web surface patterns to the generation of various relations.

We proposed another approach for unsupervised relation classification which was named multi-view co-clustering in Chapter 5. With a novel view on integrating linguistic analysis on local text with Web frequent information, we propose a multi-view co-clustering approach for semantic relation extraction. One is feature clustering by automatically learning clustering functions for Web features, linguistic features simultaneously based on a subset of entity pairs. The other is relation clustering, using the feature clustering functions to define learning function for relation extraction. Our experiments demonstrate the superiority of our clustering approach comparing with several state-of-the-art clustering methods.

In Chapter 6, we further studied how to use the clustering results for seeding a semi-supervised relation extraction system for bootstrapping a high-recall relation extraction process. We propose a multi-view learning approach for bootstrapping relationships between entities with the complementary between the Web view and linguistic view. On the one hand, from the linguistic view, linguistic features are generated from linguistic parsing on Wikipedia texts by abstracting away from different surface realizations of semantic relations. On the other hand, Web features are extracted from the Web corpus to provide frequency information for relation extraction. Experimental evaluation on a relational dataset demonstrates that linguistic analysis on Wikipedia texts and Web collective information reveal different aspects of the nature of entity-related semantic relationships. It also shows that our multi-view learning method considerably boosts the performance comparing to learning with only one view of features, with the weaknesses of one view complement the strengths of the other.

7.2 Future Research Directions

Representing natural language text in a machine understandable format will remain a challenging and interesting area of research for natural language understanding re-

searchers for the years to come. Towards deep semantic parsing for mapping a natural-language sentence into a formal representation of its meaning with good performance, we are still facing the challenges of define a universal set of semantic relations covering various types of semantic relationships between entities, and methods of automatically constructing structured format using the relation set.

Relation extraction between concepts from Wikipedia is another interesting area of research for researcher in both natural language processing and data mining areas in the further years. The semantic repository with relations and concepts will be used to different applications such as Semantic Searching or Social Network Construction. Preliminarily, stemmed from repository we constructed, one of the future work is to develop a Semantic Search Engine for Wikipedia based on the expected knowledge base. Moveover, the integration of information available on the web and linguistic techniques will be a useful idea for other information extraction tasks such as text summarization, named entity disambiguation.

Wikipedia is the world's largest collaboratively edited source of encyclopedic knowledge in different languages. Wikipedia articles consist mostly of free text, but also contain different types of structured information, such as Infobox templates, categorization information, images, geo-coordinates, and links to external Web pages and links across different language editions of Wikipedia. Another important feature of Wikipedia is the presence of parallel articles in different languages. Certain pages represent direct translations as multilingual users build and maintain a parallel corpus in different languages. A future direction will be how to extend our system to extract conceptual and relational information from other language articles such as Japanese and Chinese articles in Wikipedia database with information from the Web. There is an expectation to use Multilingual, Cross-Lingual and transfer learning technologies to make existing knowledge in one language (such as English) to provide assistance to gain knowledge in another language (Japanese). Moreover, knowledge extracted from resources in different languages (English, Japanese and Chinese) is expected to be integrated into a united format.

Appendix A

CDL relation list

No	Rel	Definition	Example
1	agt	(agent) indicates a thing in focus that initiates an action	John breaks the door
2	and	(conjunction) indicates a partner to have conjunctive relation to	He is singing and dancing
3	aoj	(thing with attribute) indicates a thing that is in s state or has an attribute	Skiiing is nice. I have a pen.
4	bas	(basis) indicates a thing used as the basis (standard) of comparison	John is more quiet than shy.
5	ben	(beneficiary) indicates an indirectly related beneficiary or victim of an event or state	It is good for John to...
6	cag	(co-agent) indicates a thing not in focus that initiates an implicit event that is done in parallel	To walk with John
7	cao	(co-thing with attribute) indicates a thing not in focus that is in a parallel state	be with you
8	cnt	(content) indicates the content of a concept	a language generator ”deconverter”...
9	cob	(affected co-thing) indicates a thing that is directly affected by an implicit event done in parallel or an implicit state in parallel	John was injured in the accident with his friends
10	con	(condition) indicates a non-focused event or state that conditions a focused event or state	If you are tired, we will go straight home

No	Rel	Definition	Example
11	coo	(effected co-thing) indicates a co-occurrent event or state for a focused event or state	... was crying while running
12	dur	(duration) indicates a period of time during which an event occurs or a state exists	...work nine hours (a day)
13	equ	(effected co-thing) indicates an equivalent concept	the deconverter (a language generator)
14	fmt	(range/from-to) indicates a range between two things	the alphabets from a to z
15	frm	(origin) indicates an initial state of a thing or a thing initially associated with the focused thing	a visitor from Japan
16	gol	(goal/final state) indicates a final state of object or a thing finally associated with the object of an event	the lights changed from green to red
17	icl	(included/a kind of) indicates an upper concept or a more general concept	a bird is a (kind of) animal
18	ins	(instrument) indicates an instrument to carry out an event	look at stars through a telescope
19	int	(intersection) indicates all common instances to have with a partner concept	an intersection of tableware and cookware
20	iof	(an instance of) indicates a class concept that an instance belongs to	Tokyo is a city in Japan
21	man	(manner) indicates a way to carry out an event or the characteristics of a state	move quickly I often visit him.
22	met	(method/means) indicates a means to carry out an event	...solve...with dynamics
23	mod	(modification) indicates a thing that restricts a focused thing	the whole story a master plan
24	nam	(name) indicates a name of a thing	his son "Hikari"
25	obj	(affected thing) indicates a thing in focus that is directly affected by an event or state	the sugar melts into... I have a pen.

No	Rel	Definition	Example
26	opl	(affected place) indicates a place in focus affected by an event	...pat...on shoulder
27	or	(disjunction) indicates a partner to have disjunctive relation to	Will you stay or leave?
28	per	(proportion/rate/distribution) indicates a basis or unit of proportion, rate or distribution	eight hours a day
29	plc	(place) indicates a place where an event occurs, or a state that is true, or a thing that exists	...cook...in the kitchen
30	plf	(initial place) indicates a place where an event begins or a state that becomes true	traveling from Tokyo
31	plt	(final place) indicates a place where an event ends or a state that becomes false	to travel to Boston
32	pof	(part of) indicate a concept of which a focused thing is a part	the preamble of a document
33	pos	(possessor) indicates the possessor of a thing	John's dog
34	ptn	(partner) indicates an indispensable non-focused initiator of an action	...compete with John
35	pur	(purpose) indicates the purpose or objective of an agent of an event or the purpose of a thing that exists	...come to see you
36	qua	(quantity) indicates the quantity of a thing or unit	Two cups of coffee
37	rsn	(reason) indicates a reason why an event or a state happens	... didn't go because of the rain
38	scn	(scene) indicates a scene where an event occurs, or state is true, or a thing exists	...win a prize in a contest
39	seq	(sequence) indicates a prior event or state of a focused event or state	It was green and then red.
40	src	(source/initial state) indicates the initial state of an object or thing initially associated with the object of an event	The lights changed from green to red.

No	Rel	Definition	Example
41	tim	(time) indicates the time an event occurs or a state is true	...leave on Tuesday
42	tmf	(initial time) indicates the time an event starts or a state becomes true	...work from morning to [till] night
43	tmt	(final time) indicates a time an event ends or a state becomes false	...be full till tomorrow
44	to	(destination) indicates a final state of a thing or a final thing (destination) associated with the focused thing	a train for London
45	via	(an intermediate place or state) indicates an intermediate place or state of an event	...bike...through the Alps

Bibliography

[1] E. Agichtein and L. Gravano. *Snowball: Extracting relations from large plain-text collections.* In Proceedings of the Fifth ACM International Conference on Digital Libraries, 2000.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. G. Ives. *Dbpedia: A nucleus for a Web of open data.* In Proceedings of ISWC-2007.

[3] C.F. Baker, C.J. Fillmore, and J.B. Lowe. The Berkeley FrameNet Project. *In Proc. 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL-98)*, 1998.

[4] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead and O. Etzioni. Open information extraction from the Web. *In Proceedings of IJCAI-2007.*

[5] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American, 2001.*

[6] A. Blum and T. Mitchell. *Combining labeled and unlabeled data with co-training.* In proceedings of COLT-1998.

[7] Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka. Measuring Semantic Similarity between Words Using Web Search Engines. *In Proceedings of WWW-2007.*

[8] D. Bollegala, Y. Matsuo and M. Ishizuka. 2009. *An Integrated Approach to Measuring the Similarity between Implicit Semantic Relations from the Web.* In Proceedings of WWW-2009.

[9] S. Brin. Extracting patterns and relations from the world wide web. *In Selected papers from the International Workshop on the WWW and Databases, London, UK, 1999. Springer.*

[10] S. Buchholz, C.J. Fillmore, and E. Marsi, CoNLL-X shared task on Multilingual Dependency Parsing. *In Proc. Tenth Conference on Natural Language Learning (COLING-X-06)*, 2006.

[11] Razvan C. Bunescu and Raymond J. Mooney. *A shortest path dependency kernel for relation extraction.* In Proceedings of HLT/EMLNP-2005.

[12] X. Carreras and L. Màrquez. Introduction to the CoNLL- 2004 shared task: Semantic role labeling. *In Proceedings of CoNLL 2004 Shared Task.*, 2004.

[13] X. Carreras and L. Màrquez. Introduction to the CoNLL-2005 shared task: Semantic role labeling. *In Proc. Ninth Conference on Computational Natural Language Learning (CoNLL-05)*, 2005.

[14] Jinxiu Chen, Donghong Ji, Chew Lim Tan and Zhengyu Niu. *Unsupervised Feature Selection for Relation Extraction.* In Proceedings of IJCNLP-2005.

[15] J. Chen, D. Ji, C.L. Tan, and Z. Niu. *Unsupervised Feature Selection for Relation Extraction.* In Proceedings of EMNLP-2006.

[16] P. Cimiano, M. Erdmann, and G. Ladwig. Corpus-based Pattern Induction for a Knowledge-based Question Answering Approach. *In Proc. IEEE International Conference on Semantic Computing (ICSC-07)*, 2007.

[17] C. Christoudias, R. Urtasun, and T. Darrell. *Multi-view learning in the presence of view disagreement.* In Proceedings of UAI-2008.

[18] T. M. Cover, J. A. Thomas. *Elements of information theory.* Wiley-Interscience.

[19] M. Collins and Y. Singer. *Unsupervised models for named entity classification.* In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.

[20] Aron Culotta and Jeffrey Sorensen. *Dependency tree kernels for relation extraction.* In Proceedings of the ACL-2004.

[21] A. Culotta, A. McCallum and J. Betz *Integrating probabilistic extraction models and data mining to discover relations and patterns in text* In Proceedings of the HLT-NAACL-2006.

[22] D. Dai, Q. Yang, G. Xue and Y. Yu. *Self-taught Clustering.* In Proceedings of the ICML-2008.

[23] D. Davidov, A. Rappoport and M. Koppel. *Fully unsupervised discovery of concept-specific relationships by Web mining.* In Proceedings of ACL-2007.

[24] D. Davidov and A. Rappoport. *Classification of relationships between nominals using pattern clusters.* In Proceedings of ACL-2008.

[25] V. R. de Sa. *Spectral clustring with two views.* In ICML Workshop on Learning with Multiple Views, 2005.

[26] I. S. Dhillon. *Co-clustering documents and words using bipartite spectral graph partitioning.* In Knowledge Discovery and Data Mining, pages 269C274, 2001.

[27] I. S. Dhillon, S. Mallela, and D. S. Modha. *Information-theoretic co-clustering.* In Proceedings of KDD-2003.

[28] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. *Web-scale information extraction in knowitall: (preliminary results).* In Proceedings of the International Conference on World Wide Web (WWW), New York, NY, USA, 2004. ACM.

[29] O. Etzioni, M. Banko and M. J. Cafarella. *Machine reading.* In Proceedings of AAAI-2006.

[30] Wei Fan, Kun Zhang, Hong Cheng, Jing Gao, Xifeng Yan, Jiawei Han, Philip S. Yu and Olivier Verscheure. *Direct Mining of Discriminative and Essential Frequent Patterns via Model-based Search Tree.* In Proceedings of KDD-2008.

[31] C. Fellbaum. WordNet: An electronic lexical database. *Cambridge, MA: MIT Press*, 1998.

[32] C.J. Fillmore. Frame semantics and the nature of language. *In Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech, volume 280, pages 20C32.*, 1976.

[33] C.J. Fillmore and C.F. Baker FrameNet: Frame semantics meets the corpus. *In Poster presentation, 74th Annual Meeting of the Linguistic Society of America,* 2000.

[34] Evgeniy Gabrilovich and Shaul Markovitch. *Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge.* In Proceedings of AAAI-2006.

[35] H. Gaifman. *Dependency systems and phrase-structure systems.* Information and Control, 8:304C337, 1965

[36] D. Gildea and D. Jurafsky. Automatic Labeling of Semantic Roles. *Computational Linguistics*, vol. 28, no. 3, pp. 245-288, 2002

[37] Jim Giles. *Internet encyclopaedias go head to head.* Nature 438:900C901.

[38] Ralph Grishman. *Information extraction: Techniques and challenges.* In Information Extraction (International Summer School SCIE-97). Springer-Verlag, 1997.

[39] J. Hajič, M. Ciaramita, R. Johansson, et al.. *The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages.* Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task, 2009

[40] D. Hays. *Dependency theory: A formalism and some observations.* Language, 40:511C525, 1964

[41] S. Harabagiu, C.A. Bejan, and P. Morarescu. Shallow semantics for relation extraction. *In Proc. Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, 2005.

[42] Z. Harris. 1954. *Distributional structure.* Word, 10:146C162, 1954.

[43] Takaaki Hasegawa, Satoshi Sekine and Ralph Grishman. *Discovering Relations among Named Entities from Large Corpora.* In Proceedings of ACL-2004.

[44] D. Haussler *Convolution Kernels on Discrete Structures.* Technical Report UCS-CRL-99-10, University of California, Santa Cruz, 1999

[45] R. Hudson. *Word Grammar.* Blackwell, 1984.

[46] T. Joachims. Text Categorization with Support Vector Machine: learning with many relevant features. *In Proc. European Conference on Machine Learning (ECML-98)*, 1998.

[47] Nanda Kambhatla. *Combining lexical, syntactic and semantic features with maximum entropy models.* In Proceedings of ACL-2004.

[48] B. Levin. English Verb Classes and Alternation: A Preliminary Investigation. *The University of Chicago Press*, 1993.

[49] K. Litkowski. Senseval-3 task automatic labeling of semantic roles. *In Proc. Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, (Senseval-3)*, 2004.

[50] B. Long, P. S. Yu, and Z. Zhang. *A general model for multiple view unsupervised learning.* In Proceedings of SDM-2008.

[51] F. Manola and E. Miller. *Resource description framework (rdf) primer.* W3C Recommendation, Feb 2004. http://www.w3.org/TR/rdf-primer/.

[52] M.Marcus, G. Kim,M.Marcinkiewicz, R.Mac-Intyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. *The Penn treebank: Annotating predicate argument structure.* In Proc. of the Workshop on Human Language Technology (HLT), 1994

[53] R. McDonald, K. Lerman, and F. Pereira. Multilingual dependency analysis with a two-stage discriminative parser. *In Proc. Tenth Conference on Natural Language Learning (CoNLL-06)*, 2006.

[54] R. McDonald, K. Lerman, and F. Pereira. Dependency Syntax: Theory and Practice. *The SUNY Press, Albany, N.Y.*, 1988.

[55] A. Meyers, R. Reeves, C. Macleod, et al. Annotating Noun Argument Structure for NomBank. *In Proc. fourth international conference on Language Resources and Evaluation (LREC-04)*, 2004.

[56] M. Minsky. Semantic Information Processing. *MIT Press, Cambridge, MA.*, 1969.

[57] S. Miller, H. Fox, L. Ramshaw, and R. Weischedel. A novel use of statistical parsing to extract information from text. *In 6th Applied Natural Language Processing Conference*, 2000.

[58] J. Mori, T. Tsujishita, Y. Matsuo, and M. Ishizuka. *Extracting Relations in Social Networks from the Web using Similarity between Collective Contexts.* In Proceedings of ISWC-2006.

[59] P. Nakov, and M. A. Hearst. *Solving Relational Similarity Problems Using the Web as a Corpus.* In Proceedings of ACL-2008.

[60] S. Narayanan, S. Harabagiu. Question answering based on semantic structures. *In Proc. 20th International Conference on Computational Linguistics (COLING-04)*, 2004.

[61] Dat P.T. Nguyen, Yutaka Matsuo and Mitsuru Ishizuka. *Relation extraction from Wikipedia using subtree mining.* In Proceedings of AAAI-2007.

[62] X. Ni, Z. Lu, X. Quan, L. Wenyin, and B. Hua. *Short Text Clustering for Search Results.* In Proceedings of APWeb-WAIM-2009.

[63] K. Nigam and R. Ghani. *Analyzing the effectiveness and applicability of cotraining.* In Workshop on Information and Knowledge Management, 2000.

[64] J. Nivre. Constraints on non-projective dependency parsing. *In Proc. 11st Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

[65] M. Palmer, D. Gildea, and P. Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, vol. 31, no. 1, 2005.

[66] Patrick Pantel and Marco Pennacchiotti. *Espresso: Leveraging generic patterns for automatically harvesting semantic relations.* In Proceedings of ACL-2006.

[67] S. P. Ponzetto and M. Strube. *Deriving a large-scale taxonomy from Wikipedia.* In Proceedings of AAAI-2007.

[68] Benjamin Rosenfeld and Ronen Feldman. *URES: an Unsupervised Web Relation Extraction System.* In Proceedings of COLING/ACL-2006.

[69] Benjamin Rosenfeld and Ronen Feldman. *Clustering for Unsupervised Relation Identification.* In Proceedings of CIKM-2007.

[70] S. Sekine. *On-demand information extraction.* In Proceedings of COLING-2006.

[71] Y. Shinyama and S. Sekine. *Preemptive Information Extraction using Unrestricted Relation Discovery.* In Proceedings of HLT-NAACL-2006.

[72] V. Sindhwani, P. Niyogi, and M. Belkin. *A coregularization approach to semi-supervised learning with multiple views.* In Proceedings of ICML-2005.

[73] D. Sleator and D. Temperley. *Parsing English with a link grammar.* In Proc. of the 3rd Intern. Workshop on Parsing Technologies (IWPT), 1993

[74] F. M. Suchanek, G. Kasneci, G. Weikum. *YAGO: A Core of Semantic Knowledge.* In Proceedings of WWW-2007.

[75] F. M. Suchanek, G. Kasneci, G. Weikum. *YAGO: A Large Ontology from Wikipedia and WordNet.*. Elsevier Journal of Web Semantics, 2008.

[76] F. M. Suchanek, M. Sozio, G. Weikum. *SOFIE: A Self-Organizing Framework for Information Extraction.* In Proceedings of WWW-2009.

[77] P. Tapanainen and T. Jarvinen. A non-projective dependency parser. *In Proc. 5th Applied Natural Language Processing Conference (ANLP-97)*, 1997.

[78] L. Tesnière. *Elèment de syntaxe structurale.* Klincksieck, Paris.

[79] Peter D. Turney. *Expressing implicit semantic relations without supervision.* In Proceedings of ACL-2006.

[80] H. Uchida. *Universal Networking Language (UNL) Specifications Version 2005.* UNL Center of UNDL Foundation, 2005. http://www.undl.org/unlsys/unl/unl2005/.

[81] Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller and Rudi Studer. *Semantic wikipedia.* In Proceedings of WWW-2006.

[82] G. Wang, Y. Yu, H. Zhu. *Pore: Positive-only relation extraction from Wikipedia text.* In Proceedings of ISWC-2007.

[83] T. Winograd. *Understanding natural language.* Cognitive Psychology, 3(1):1C191. Reprinted as a book by Academic Press, 1972.

[84] F. Wu and D. S. Weld. *Autonomously semantifying Wikipedia.* In Proceedings of CIKM-2007.

[85] F. Wu and D. S. Weld. *Automatically refining the Wikipedia infobox ontology.* In Proceedings of WWW-2008.

[86] T. Yokoi, H. Yasuhara, H. Uchida, et al. CDL (Concept Description Language): A Common Language for Semantic Computing. *In WWW2005 Workshop on the Semantic Computing Initiative (SeC2005)*, 2005

[87] Mohammed J. Zaki. 2002. *Efficiently mining frequent trees in a forest.* In Proceedings of SIGKDD-2002.

[88] D. Zelenko, C. Aone, and A. Richardella. *Kernel Methods for Relation Extraction.* Journal of Machine Learning Research. 3(Feb):1083-1106, 2003.

[89] Min Zhang, Jie Zhang, Jian Su and Guodong Zhou. 2006. *A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features.* In Proceedings of ACL-2006.

[90] S. Zhao and R. Grishman. *Extracting relations with integrated information using kernel methods.* In Proceedings of ACL-2005.

[91] G. Zhou, J. Su, J. Zhang, and M. Zhang. *Exploring various knowledge in relation extraction.* In Proceedings of ACL-2005.

[92] G. Zhou, M. Zhang, D. Ji and Q. Zhu. *Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information.* In Proceedings of EMNLP-2007.

[93] J. Zhu, Z. Nie, X. Liu, B. Zhang and J.R. Wen *StatSnowball: a Statistical Approach to Extracting Entity Relationships.* In Proceedings of WWW-2009

Publication List

Journal Papers

Yulan Yan, Yutaka Matsuo, Mitsuru Ishizuka. Automatically Relation Extraction from Wikipedia Texts by Exploring Web Features and Linguistic Features. Submitted to Information Processing & Management. December, 2009

Yulan Yan, Yutaka Matsuo, Mitsuru Ishizuka. A New Shallow Semantic Parser for Describing the Concept Structure of Text. Int'l Journal of Semantic Computing, 2009.

Yulan Yan, Keqing He, Jin Liu. A Finite-State-Based Model Transformation method. Journal of Computer Engineering, 2006.

International Conference Papers

Yulan Yan, Haibo Li, Yutaka Matsuo, Mitsuru Ishizuka. Multi-View Bootstrapping for Relation Extraction by Exploring Web Features and Linguistic Features. To appear in the 11th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2010)

Yulan Yan, Haibo Li, Yutaka Matsuo, Zhenglu Yang, Mitsuru Ishizuka. Multi-View Clustering with Web and Linguistic Features for Relation Extraction. To appear in the 12th International Asia-Pacific Web Conference (APWeb-2010)

Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang and Mitsuru Ishizuka. Unsupervised Relation Extraction by Mining Wikipedia Texts Using Information from the Web. The Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL2009), 2009.

Yulan Yan, Yutaka Matsuo, Mitsuru Ishizuka. An Integrated Approach to Extracting Relations from Wikipedia Texts. In the Proceedings of WWW2009 Workshop on Content Analysis in Web 2.0, 2009

Yulan Yan, Yutaka Matsuo, Mitsuru Ishizuka, Toshio Yokoi. Relation Classification for Semantic Structure Annotation of Text. In the Proceedings of 2008 IEEE/WIC/ACM Int'l Conf. on Web Intelligence (WI'08), Sydney, Australia, 2008

Yulan Yan, Yutaka Matsuo, Mitsuru Ishizuka, Toshio Yokoi. Annotating an Extension Layer of Semantic Structure for Natural Language Text. In the Proceedings of IEEE Int'l Conf. on Semantic Computing (ICSC2008), Santa Clara, CA, USA, 2008

Domestic Conference Papers

Yulan Yan, Yutaka Matsuo, Mitsuru Ishizuka. Transfer learning-based co-clustering over Linguistic and Web features. The 23rd JSAI. 2009

Yulan Yan, Yutaka Matsuo, Mitsuru Ishizuka, Toshio Yokoi. Annotating Semantic Structure of Web Text based on CDL.nl. FIT2008, 2008